GTAPinGAMS with a GUI

- Wolfgang Britz, Institute for Food Resource Economics –

- October 2015 -

(matches rev. 313 of https://svn1.agp.uni-bonn.de/svn/Gtap8InGams)

Inhalt

Background ... 3
Why an implementation in GAMS? ... 3
Scalability, Modularity and Extensions ... 3
Flexible nesting ... 5
GUI ... 8
Working with the GUI ... 10
Getting the model code and setting up the GUI ... 10
SVN checkout ... 10
Starting the GUI ... 13
First steps ... 15
Making the interface working with GAMS ... 17
Data base generation ... 20
Simple deletion ... 20
Rebalancing ... 21
Special treatment for specific regions and sectors ... 22
Model size and solution behavior ... 23
Inspecting the resulting data base ... 25
Pre- or post-model aggregation ... 26
Scenario runs ... 27
Defining the scenario ... 27
Running the scenario ... 30
Output options ... 31
Model structure, parameterization and factor mobility ... 32
Modules and parameters ... 32
Numeraires ... 33
Closures ... 33
Algorithm ... 35
GAMS ... 37

Inspecting the results 37
Notes for GEMPACK users or users familiar with other CGE models 39
Relative difference for all variables as in runGTAP 40
Trouble shooting with the viewer 42
Overview on existing views for exploitation 43
Sensitivity analysis 53
Controls for sensitivity experiments 55
Entropy estimator 55
Drawing systematic experiments for other elements of the simulation 55
Parallel solves and analyzing the results 56
Getting an overview on the code 61
HTML documentation 61
Equation and variable viewer (similar AnalyzeGE) 63
Technical documentation 65
Overview on directory structure 65
GUI overview 66
Processes 67
Post-model processing 68
Exploitation and flexible aggregation 70
References 73

Background

Why an implementation in GAMS?

GTAP is certainly the far most widespread international data base for CGE modeling. The data base was so far only distributed with a default CGE model template in GEMPACK, called the standard GTAP model, as well as with many useful GEMPACK based utilities. GEMPACK as a special package for CGEs has certainly distinct advantages compared to less specialized languages. Some users might however prefer GAMS when working with GTAP. Indeed, outside the CGE community, GAMS seems much more common than GEMPACK as an Algebraic Modelling Language, e.g. in Agricultural Economics (Britz and Kallrath 2012). In 2015, the Center for Global Trade Analysis (GTAP) decided to release in parallel to the standard GTAP model in GEMPACK a version coded in GAMS (van der Mensbrugghe 2015). Intensive testing shows that both versions produce the same results.

CGE models realized in GAMS sharing basic elements of the standard GTAP model while adding extensions in several directions are widely available (e.g. GTAPinGAMS by Tom Rutherford, GLOBE by Scott McDonald and Karen Thierfelder or ENVISAGE by Dominique van der Mensbrugghe). However, these models do not provide an exact replication of the standard GTAP Model. The newly released replica of the standard GTAP model in GAMS is not only interesting for research, but also provides together with the GTAP data base an excellent starting point either for teaching or own projects of students, especially in the context of course programs already using GAMS. It offers additional some extensions compared to the standard GTAP model as discussed next, inspired by features of existing CGEs.

Scalability, Modularity and Extensions

As the standard GTAP model, the GAMS version is highly scalable in the sense that it can work with differently sized, but identically structured data bases. The GAMS code comprises a pre-solve algorithm which should speed up solution of larger models with more than 300.000 non-empty transactions in the global SAM. The code was intensively tested for numerical stability and acceptable solution time when applied to more dis-aggregated versions of the GTAP data base.

As mentioned above, the GTAP in GAMS code includes and supports extensions beyond the GTAP standard model. Using these extensions is partly driven by the underlying SAM structure, such as allowing for a multi-product make matrix to support non-diagonal relations between production activities and commodities consumed. Other extensions are not driven by the structure of the data base, but switched one depending on the parameterization of the system, such as allowing for non-infinite CET between domestic sales and exports, and between export flows. Similarly, non-diagonal make matrices can be applied in conjunction with a CET nest to steer output composition of multi-output sectors and a CES nest to determine consumption shares in case several sectors produce the same commodity. Many of these extensions are inherited from ENVISAGE, however not all ENVISAGE features are found in the code. The code also support already a range of different closures e.g. for foreign savings, private and government consumption.

Furthermore, the code structure has a modular design which facilitates introduction of new model features to the GAMS code. So far, implementations of GTAP-AGR, GTAP-E, GTAP-AEZ and CO2 emissions accounting and taxation are available as additional modules, as well as the production

nesting of the GTAP-E model[1]. It should however be noted that these extensions are partly not set-up as a prefect replica of the GEMPACK implementation and not tested for replication of results obtained with the corresponding GEMPACK code.

An important feature of the model's structure is a set of equations which allow the introduction of CES-nests comprised of factors, intermediates demand and sub-nests such that even complex nestings in the production function can be introduced via cross-set definitions without additional programming work in the model's code. A similar generic implementation is provided for factor supply based on nested CET-functions and for CES subnests under final demand. These features are discussed below in more detail.

The basic application mode of the model, according to the GTAP standard model, is that of a comparative-static global model of a barter economy, i.e. with fixed exchange rates. Alternatively, the model can also be run in recursive-dynamic fashion, again drawing on ENVISAGE. A formal comparison to the features of dynamic GTAP model Gdyn is pending. Equally, a single regional model can be directly derived from the equation system, and the single region to run selected form the GUI. We also allow for a version where exchange rates are flexible.

Use of the options, extensions and modules discussed above does not require additional coding efforts; they can be directly switched activated from the interface or by using non-default parameter files. The set-up of the code should render it relatively straightforward to implement in a modular fashion additional module.

Table 1: Modules and extensions

Module	Status	Remarks
Data filter	Tested, not used with GEMPACK based model	Optionally removes small transactions from SAMS / trade matrices while maintaining closely important totals. Thought to support model applications with highly dis-aggregated data bases.
GTAP-Standard	Tested for exact replication	With extensions from ENVISAGE such as non-diagonal make matrix, CET on export side
Completely flexible nesting of production functions	Tested, based on set-definitions	Should allow to quickly generate variants of the standard GTAP model currently available which differ in nesting of factors / intermediate
Completely flexible nesting for factor supply	Tested, based on set-definitions	Should allow to quickly generate variants of the standard GTAP model currently available which use nested CET structure to describe factor supply
Completely flexible	Tested, based on set-definitions,	Should allow to quickly generate variants

[1] The code makes a differentiation between extensions which change the nesting of the production function or introduce nested CET structures for factor supply. These extensions only require set-definitions and substitution respectively transformation elasticities, but no changes in the equation structure of the model or is post-processing. In opposite to that, the implementation of a what is called a module consists typically of two files: one file which declares the additional symbols (parameters, variables, equations), found in "model.gms", and a file which calibrates the related parameters for the benchmark, found in "cal.gms". Potentially, there are also additional statements in the post-model reporting part.

of sub-nests under final demand	not available for the trade margins	of the standard GTAP model currently available which use CES-subnests under the top-level final demand equation
GTAP-AEZ	Operational, not tested for exact replications	With land supply elasticities for natural land cover, differentiated by land cover
GTAP-AGR	Operational, not tested for exact replications	Applicable also to regional dis-aggregation different from original GEMPACK implementation, uses the flexible nesting approach
GTAP-E	Operational, not tested for exact replications	Based on flexible nesting approach
Aggregate Armington aggregator for intermediate demand	Operational, not part of standard GTAP model or variants thereof available from GTAP	Domestic and import shares for intermediate demand and related tax rates are not longer sector specific, removes a large share of equations
Post-model reporting	Could be done in GEMPACK	Generates SAM like structure, calculates world totals, regional and sectoral totals based on additional GTAP agg file, welfare decomposition etc., feeds into GUI exploitation tools
Single region mode	Operational, not available in standard GTAP model	Fixes import prices and let export demand react to lower Armington nest at export destinations
Recursive dynamic mode	Operational, not tested for compliance with GDyn	

Flexible nesting

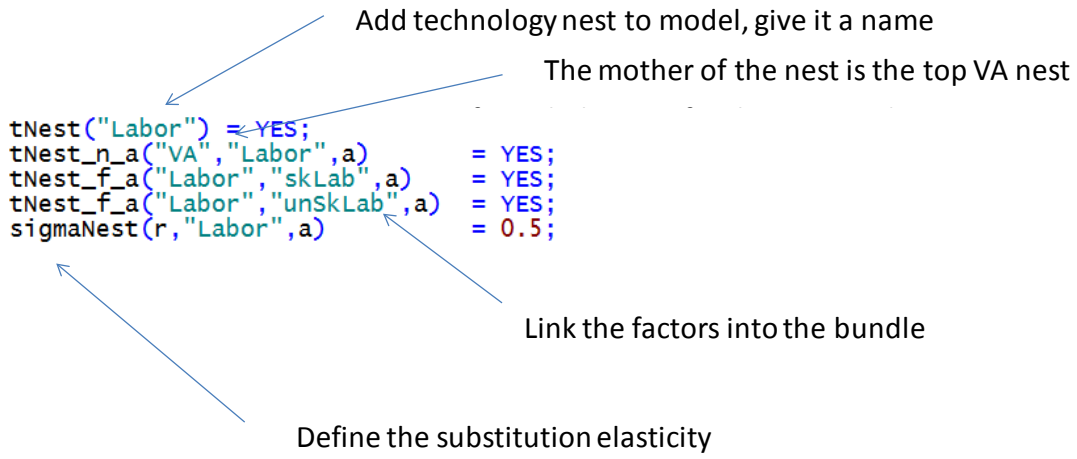
While most CGE models apply the CES functional form to depict the production function, quite some differences exist how input composites are defined. Usually, each nest is represented in the model's programming code by its own quantity and price aggregator equation. Adding or changing nests thus requires coding efforts – new variables, equations and parameters need to be defined and properly assigned. Typically, also the equations relating to the top level value added aggregator and/or the intermediate demand nest need to be adjusted if the nesting is changed.

The GAMS code underlying the GTAP in GAMS project applies a different strategy. Here, nested CES structures in the production function are represented by a generic approach where a small number of equations and matching variables handle basically all possible nesting structures. The equations for the top level VA and ND nests are already set-up to host sub-nests along with equations describing sub-nests. In the standard GTAP-implementation, such sub-nests are not present and these equations empty.

That flexible and generic nesting approach is based on sets and cross-sets in GAMS which define the lists of factors, intermediates and sub-nests comprised in a CES composite nest along with the top nest it belongs too. These nesting definitions enter the above mentioned equations in the core model and matching code dealing with parameter calibration. Hence, the user does not need to introduce additional equations in the code to use the feature – it is sufficient to provide the structure of the nesting used via set definitions and the related substitution elasticities. The code also tests for potential errors such as duplicate assignments or sub-nests not linked into another nests.

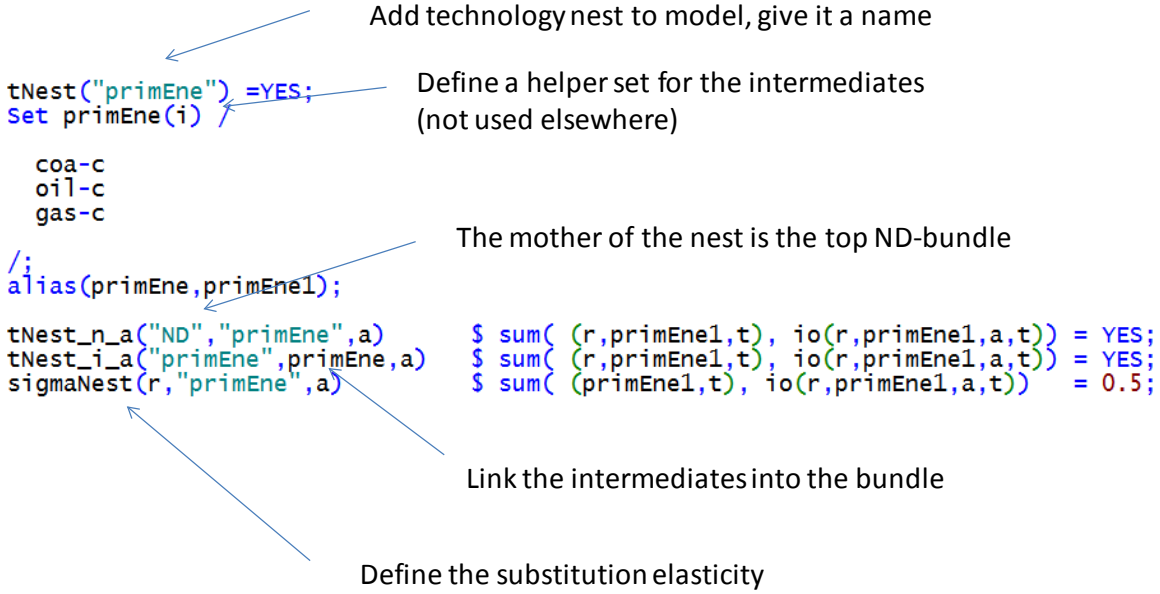
The following examples should be sufficient to show the application of that feature and demonstrate its flexibility.

1. An example of a sub-nest under the top VA nest which aggregates the two labor categories found in the GTAP 8 data base into an aggregate:


Add technology nest to model, give it a name
The mother of the nest is the top VA nest
tNest("Labor") = YES;
tNest_n_a("VA","Labor",a) = YES;
tNest_f_a("Labor","skLab",a) = YES;
tNest_f_a("Labor","unSkLab",a) = YES;
sigmaNest(r,"Labor",a) = 0.5;
Link the factors into the bundle
Define the substitution elasticity

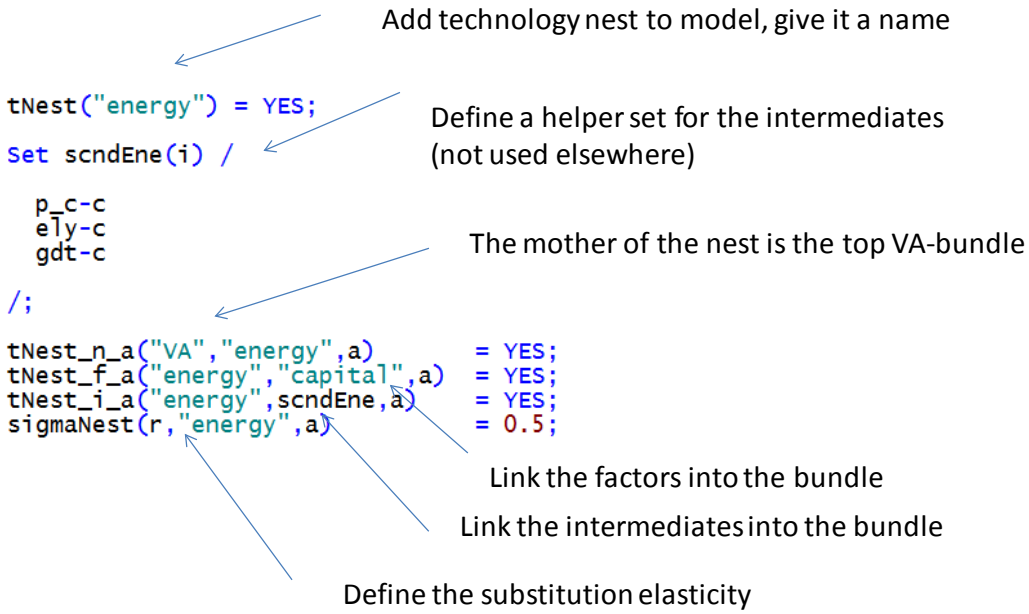
Note: The code will automatically remove the factors linked into nests from the top VA nest

2. The second example shows how to introduce a CES-composite of intermediates linked into the top ND-nest, as e.g. applied in GTAP-AGR to allow for high substitution of feed intermediates in animal production:


Add technology nest to model, give it a name
tNest("primEne") =YES;
Set primEne(i) /
coa-c
oil-c
gas-c
/;
Define a helper set for the intermediates
(not used elsewhere)
The mother of the nest is the top ND-bundle
alias(primEne,primEne1);
tNest_n_a("ND","primEne",a) $ sum((r,primEne1,t), io(r,primEne1,a,t)) = YES;
tNest_i_a("primEne",primEne,a) $ sum((r,primEne1,t), io(r,primEne1,a,t)) = YES;
sigmaNest(r,"primEne",a) $ sum((primEne1,t), io(r,primEne1,a,t)) = 0.5;
Link the intermediates into the bundle
Define the substitution elasticity

Note: The code will automatically remove intermediates linked into nests from the top ND bundle

3. The third example shows how to combine intermediates and factors into a CES composite, which in the example is a sub-nest of the top VA nest:



4. The last example reproduces the nesting of the GTAP-E model (the composite of skilled and unskilled labor is already shown above):

```
*
* --- top level nest, aggregate of capital and energy
*
  tNest("CAP+ENE") = YES;
  tNest_n_a("VA","CAP+ENE",a)      = YES;
  tNest_f_a("CAP+ENE","capital",a) = YES;
  tNest_n_a("CAP+ENE","energy",a)   = YES;
  sigmaNest(r,"CAP+ENE",a)         = 0.25;

  tNest("energy") = YES;
  tNest_i_a("energy","ele-c",a)        = YES;
  tNest_n_a("energy","non-electric",a) = YES;
  sigmaNest(r,"energy",a)              = 1.00;

  tNest("non-electric") = YES;
  tNest_i_a("non-electric","coa-c",a)    = YES;
  tNest_n_a("non-electric","non-coal",a) = YES;
  sigmaNest(r,"non-electric",a)          = 0.50;

  tNest("non-coal") = YES;
  tNest_i_a("non-coal","gas-c",a)    = YES;
  tNest_i_a("non-coal","oil-c",a)    = YES;
  tNest_i_a("non-coal","p_c-c",a)    = YES;
  sigmaNest(r,"non-coal",a)          = 1.00;
```

The nesting used in a specific model run along with the substitution elasticities are reported in a table in the exploitation tools. Equally, the code reports the quantity and price aggregators for nests and the resulting values and aggregates them up over individual sectors and over regions.

It is important to note that the post-model reporting redefines the top-level VA and ND nests such that they match the usual definition, i.e. an aggregation over primary factors and intermediates, respectively. Information about these nests in the definition used in the model can be retrieved with the separate tables showing all technology nests.

A similar generic approach is implemented for factor supply: nested CET functions can be used to supply primary factors to the production sectors. The top level nest is labeled with “xft”. The following example used in the GTAP-AGR implementation shows that approach:

```
*
* --- build nesting structure for factor supply to agr / non-agr
*
   fNest("agr")                  = YES;
   fNest_a_f("agr",agr,fm)       = YES;
   fNest_n_f("xft","agr",fm)     = YES;
   omegafNest(r,"agr",fm)        = omegaf(r,fm);

   fNest("nonAgr")               = YES;
   fNest_a_f("nonAgr",nonAgr,fm) = YES;
   fNest_n_f("xft","nonAgr",fm)  = YES;
   omegafNest(r,"nonAgr",fm)     = omegaf(r,fm);
```

Similar to the CES production nests, the post-model processing reports the structure, parameterization and simulation results in the exploitation tools.

Finally, these flexible nestings are also applicable for final demand (government, investments and households), as shown here again with the nesting used by GTAP-E:

```
dNest("energy") = YES;
dNest_i_fd("energy","ele-c",fdn)         = YES;
dNest_n_fd("energy","non-electric",fdn) = YES;
dNest_n_fd("top","energy",fdn)           = YES;
sigmaFDNest(r,"energy",fdn)              = 1.00;

dNest("non-electric") = YES;
dNest_i_fd("non-electric","coa-c",fdn)    = YES;
dNest_n_fd("non-electric","non-coal",fdn) = YES;
sigmaFDNest(r,"non-electric",fdn)          = 0.50;

dNest("non-coal") = YES;
dNest_i_fd("non-coal","gas-c",fdn)    = YES;
dNest_i_fd("non-coal","oil-c",fdn)    = YES;
dNest_i_fd("non-coal","p_c-c",fdn)    = YES;
sigmaFDNest(r,"non-coal",fdn)          = 1.00;
```

GUI

Working with a larger existing model such as GTAP can provoke serious entry barriers for newcomers, partly due to the need to learn specific model mnemonics underlying the code. That knowledge is necessary to analyze results if only model output directly produced by the modeling language is available, e.g. listings or proprietary binary result files. Equally, the sheer amount of symbols found in larger models (parameters, variables, equations, labels for regions, sectors, factors etc.) and the dimensionality of results can overwhelm beginners. That might shy them away from a fascinating research field, even if excellent model documentation is available, such as in the case of GTAP. Memorizing model mnemonics and partly structure is rather independent from the modeling language used –beginners will always have to digest a larger amount of information to make the first steps with a larger economic simulation model.

It might therefore be useful to reduce entry barriers such that students might become enthusiastic about the model’s capabilities and then take the more tedious steps to familiarize themselves with the model’s code. Therefore, the standard GTAP model in GAMS comes with a Graphical User Interface (GUI) which complements the GAMS code. The reader should however note that the GUI

linked to the GTAP in GAMS version follows a somewhat different approach compared to runGTAP, the GUI available for the GEMPACK version (see also Britz 2014 and Britz et al. 2015) or e.g. the GAMSIDE often used with economic models realized in GAMS. As a consequence, some functionalities found in runGTAP are not comprised the GUI linked to the GAMS version of the GTAP model and vice versa. The development of the GUI for the GAMS version and GAMS code for post-model processing feeding into the exploitation part of the GUI went partly in parallel with the development of the model codes to ensure seamless inter-operability. However, it is important to note that the GAMS code can be used completely independent from the GUI.

The GUI provides two major functionalities: firstly, it allows running the model while also liberating a user to carry out changes that are otherwise necessary in the GAMS code—such as to select a specific data base or shock file or to switch modules on and off. Instead, controls on the GUI allow steering the model run, thereby avoiding code edits. Secondly, the GUI offers tools to view and analyze model results. Specifically, a post-model reporting part maps back the variables into a structure similar to a SAM, and additionally stores quantities, prices, values, tax rates and tax incomes. That structure can then be viewed in a flexible exploitation tool. The post model code also provides some eventually useful aggregations, e.g. to total intermediate input, total factor input, total output, total demand and world totals and provides a welfare decomposition. The user can also provide based on an aggregation definition file generated with GTAPAgg a second, more aggregated sectoral and/or regional aggregation which is used for post-model aggregation of these results. The exploitation tools available with the GUI provide interesting features: they allows to view simultaneously results from different shocks without running additional programs, support maps and a wider range of graphics as well as basic statistics and thus might hence also be of interest to seasoned modelers. Furthermore, the GUI comprises some additional utilities: a tool similar to "EXAMINER" in GTAP which shows the linearized version of the model's equations and allows inspecting parameters and variables found in these equations. Similar to "AnalyzeGE", that tool also allows decomposing changes of the LHS of each equation to changes of the variables on the RHS. Another utility allows documenting the code in HTML pages. Finally, instead to using the GUI in interactive model, a batch mode allows to perform model runs. The batch reports these run in a HTML page and allows parallel execution of different shocks or of the same shock in different model variants.

The paper provides a hand-on guide on how to use the model in conjunction with the GUI. The latter is based on a GUI generator for GAMS projects (http://www.ilr.uni-bonn.de/agpo/staff/britz/ggig_e.htm, Britz 2010b), which is also used for some larger projects such as CAPRI (http://www.capri-model.org/dokuwiki/doku.php?id=start), but also a range of other modeling system such as the Policy Evaluation Model (PEM) by the OECD; a dynamic single farm model, an Agent Based Model of structural change in agriculture or a recursive-dynamic Hydro-economic river basin model. For those interested in a discussion about interfaces used with economic simulation models, see for example https://www.gtap.agecon.purdue.edu/resources/download/5864.pdf (Pérez Domínguez et al. 2012) or Britz et al. 2015. The GUI generator is flexible enough to also host extensions as shown by the implementations of GTAP-AEZ, GTAP-AGR and GTAP-E.

A first version of the GUI was developed for a class at Bonn University, Institute for Food and Resource Economics, in the winter term 2013/2014, building on Tom Rutherford's GTAPinGAMS version. The students had already some knowledge of GAMS and CGEs, and wanted to use a global

CGE in a project. A year later, again a class of students wanted to use a global CGE, which led to further improvements such as flexible aggregation of regions in the mapping viewer or the introduction of sensitivity analysis. In 2014, when the GTAP Center decided to release a full-fledged version of the standard GTAP model in GAMS, it was decided to adjust the interface to operate with that version. The further work on GTAP led to improvements to the GUI generator, also to the benefit of other modeling systems applying the GUI. The GUI itself is distributed for free. However, the underlying Java code is currently not open source.

Working with the GUI

Getting the model code and setting up the GUI

The code and GUI are distributed in three ways. First, a GAMS code only package can be downloaded as a zip archive from the GTAP web site. The GAMS code is identical to the one used in the GUI version, but the java binaries and other files necessary for the GUI are not included. That GAMS only package might be preferred by users wanting to use tools such as the GAMSIDE, develop their own GUI or may want to link the GTAP code into other GAMS projects. Second, a zipped archive "Gtap8InGams.zip" at ftp.agp.uni-bonn.de (userid and pwd is gtapingams) comprises both the GAMS code and the necessary files for the GUI.

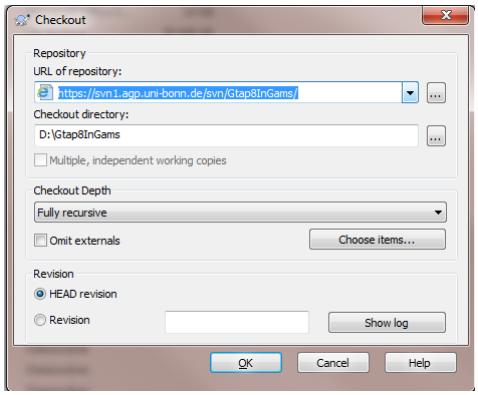
SVN checkout

The third option is to benefit from SVN, a software version system on which the model codes are hosted on the repository https://svn1.agp.uni-bonn.de/svn/gtap8ingams/. The option is especially recommended for users who both want to use the GUI and foresee also own changes to the code. In the SVN repository, the development of the standard GTAP in GAMS continues. We expect that over the next few months additional modules will be integrated into the code and clearly, bugs corrected and improvements implemented once the code is more widespread used. Using SVN allows updating more easily to bug fixed or new extensions.

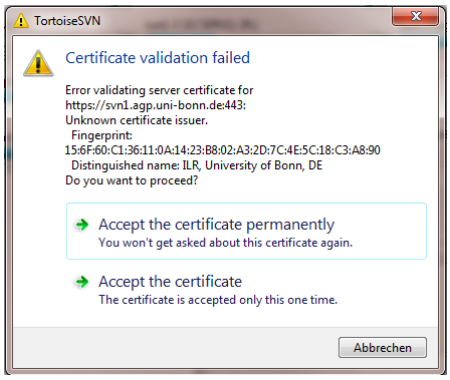
Therefore, as a first step, we recommend installing a SVN client such as TortoiseSVN (http://tortoisesvn.net/index.de.html, freeware) on your machine. Afterwards, make a checkout of the repository https://svn1.agp.uni-bonn.de/svn/Gtap8InGams/ (i.e. download the code from the server to your local machine), using the user id "gtapingams" and the pass word "gtapingams". Using SVN will allow you to keep your local copy automatically updated to code improvements, to check which changes you introduced in the code and to switch easily back to original versions of all or specific files if deemed useful, if necessary on a line by line basis.

The basic steps for using SVN to install GTAPinGAMS based on TortoiseSVN are detailed below. First, navigate in the windows explorer to the disk / directory where you want to install GTAP, and open the context menu with a right mouse click:

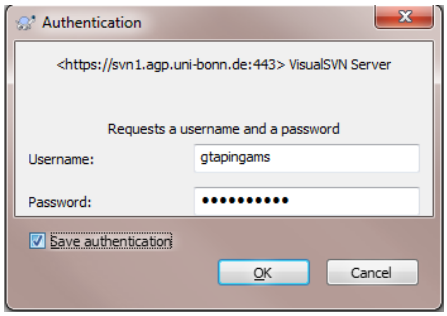
Chose, "SVN Checkout …" and copy the URL "https://svn1.agp.uni-bonn.de/svn/Gtap8InGams/" in the first field as shown below and press "OK".



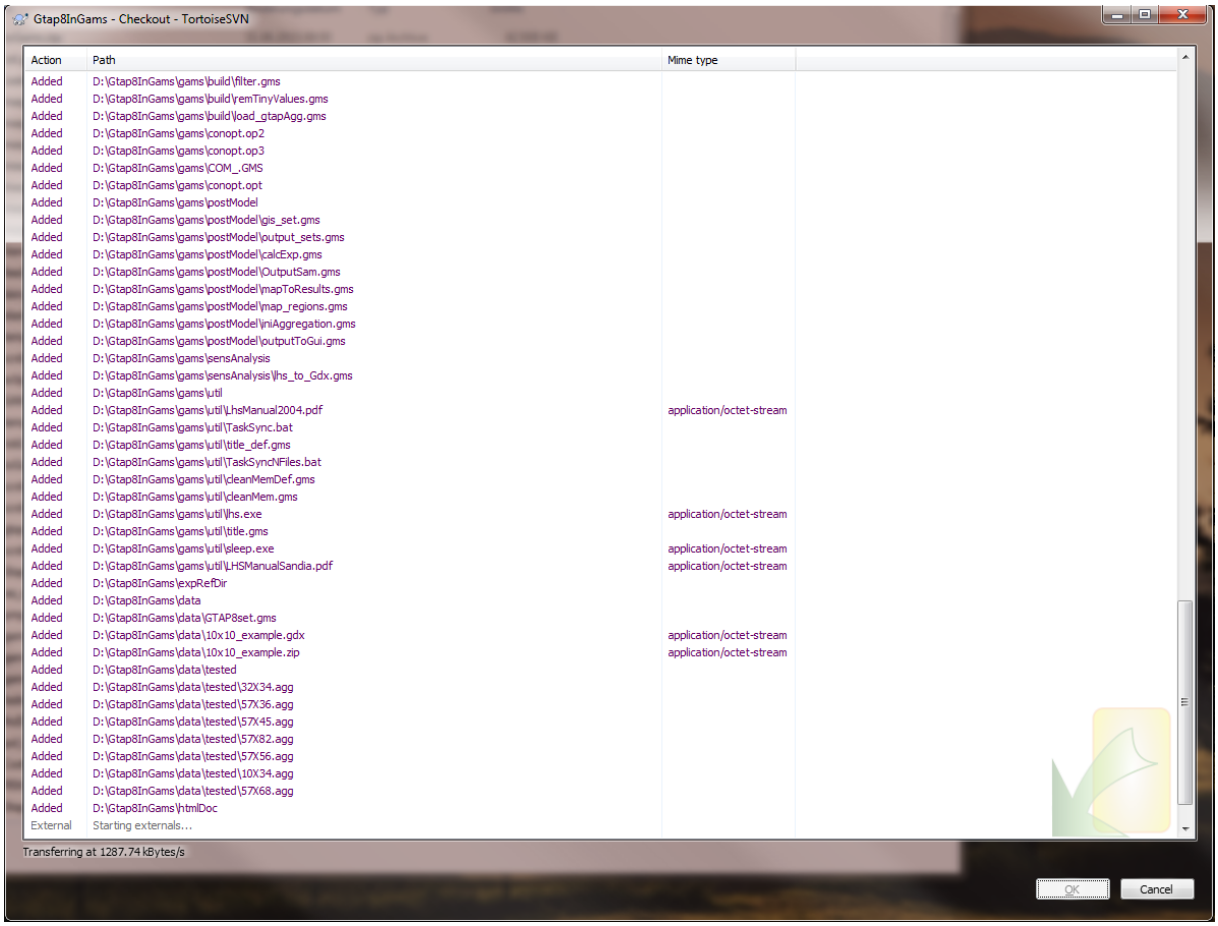
In case you get a warning such as:



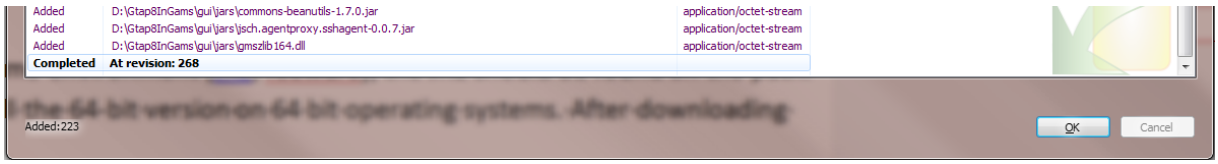
chose “Accept the certificate permanently”. In the next dialogue, enter the username and password, check “Save authentication” in the checkbox in the lower part of the dialog and press “OK”.



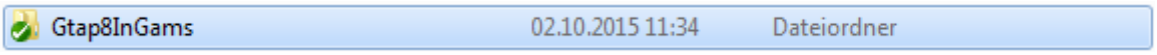
The SVN client will next download the newest release of all files to your computer which might take some time:



And should end with a “Completed At revision ..” message.



After you press “OK”, you should find the new folder on your computer:

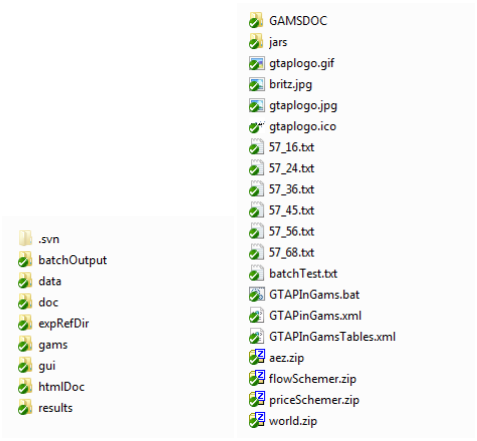


The green mark indicates that there are no local modifications to files found in the folder, i.e. that the files on your disk are identical to those just downloaded from the repository. If you introduce changes in these files, a different symbol will be shown.

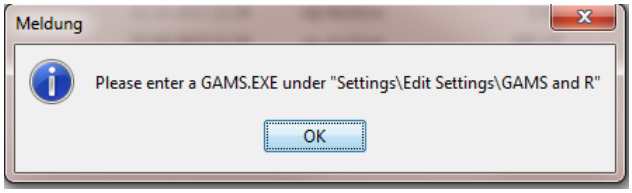
For further information on how to benefit from TortoiseSVN for your daily work with GTAP8InGAMS for other projects, please refer to the web. We would like to mention here only that you have always two versions of each file on your computer: (1) the so-called local working copy – these are the files in your normal directory with which you can work as usual and (2) a second version in a hidden data base which reflects the latest download from the repository. That allows the system to e.g. find out which files changed and to highlight changes line by line. It is important to note that you will never lose edits on your local computing when you perform “updates”, i.e. download newer versions from the central repository.

Starting the GUI

The GUI requires a Java 8 run time environment (JRE, feeware); the JRE should be found on the path. Please make sure that you install the 64-bit version of the JRE on 64 bit operating systems. After downloading the GAMS sources and other files via SVN or having unzipped the downloaded folder, navigate to the GUI folder:



and double-click in the GUI folder on the file "*GTAPinGAMS.bat*"; it should open the interface[2]. You might want to put a shortcut to that batch file on your desktop to easily start GTAPinGAMS. You first see a message which tells you that a GAMS executable is not yet registered:



You can first ignore that warning for now. After pressing "OK", you should see a program opened as below:

[2] If that does not work, open a command prompt, start from there the "GTAPinGAMS.bat" batch file and analyze the error. Most probably, JAVA is not found. In that case, either put it on the path or change the batch file such that it calls Java from the directory where it is installed.

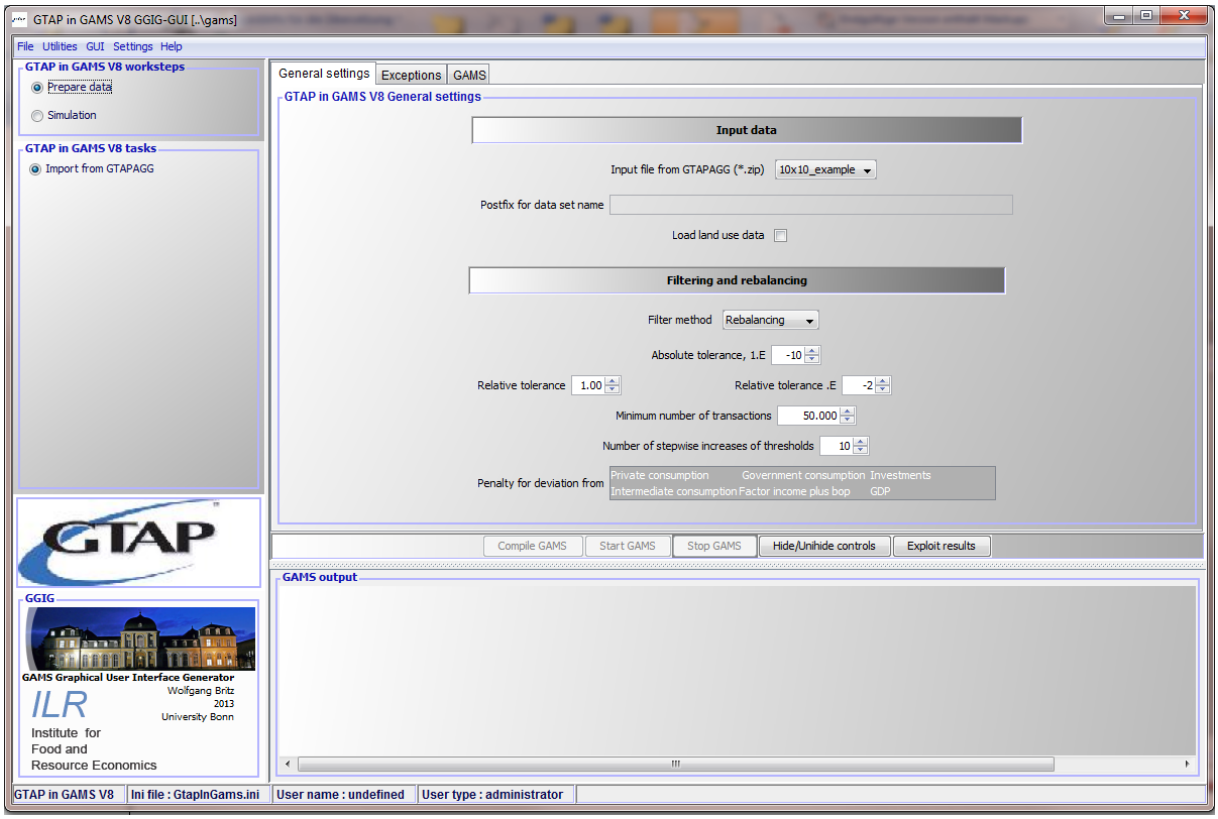


Figure: Interface at first start

The left hand side allows selecting work steps and the tasks comprised in them; the right hand side carries controls to steer these tasks. The bar in the middle comprises buttons to start the GAMS code of a task and exploit the resulting output.

Typical reasons why the ***GUI does not open*** and possible remedies are described briefly below:

- ***Java is not installed*** or not registered on the PATH. In order to test that, open a DOS prompt and type Java. If you receive a message that the command is not found, (re-)install Java and make sure that Java also updates the PATH during the installation.
- A ***wrong Java Runtime Environment*** is installed. Please first check if Java version 8 is installed and not some older version (that check is not only useful for the GUI of GTAP8inGAMS, but also generally for security reasons if one uses Java plugins in web pages). *On 64 bit Windows operation systems, a 64 bit version of Java should be used*. You can test that again in a DOS prompt by typing "java –version". The response should look similar as shown below (note the string "64-Bit"). Unfortunately, many 64 bit Windows machines use a 32 bit browser, and if you install Java from such as browser, the 32 bit version is downloaded. The link JRE points both to a 32 and 64 bit version, make sure you select the 64 bit version on a computer with a 64 bit OS. Installing a 64 bit version on a machine where the 32 bit version is already installed does no harm and makes the 64 bit version the default.

```
java version "1.8.0_25"
Java(TM) SE Runtime Environment (build 1.8.0_25-b18)
Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)
```

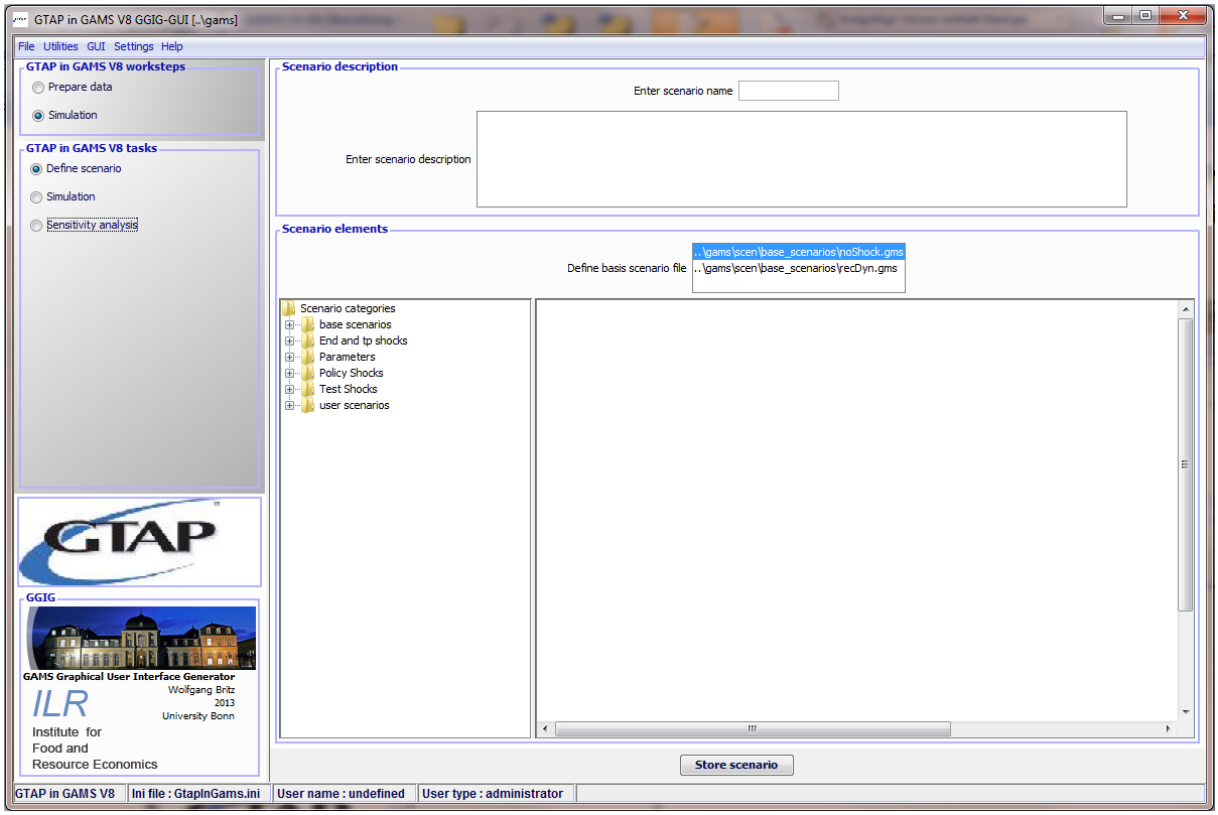
- If that still does not help, open a DOS prompt, navigate to the GUI folder and type "GtapInGams.bat" – possible errors will be shown in the window. In case you cannot find help in your team with regard to these error messages, contact the author.

General information on how to work with the GUI can be found in the GGIG user guide at http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG_user_Guide.pdf. That pfd document can also found in the folder "GUI\jars" in your installation. Please note that the hints given below in the document on using the interface are only thought as a first introduction, refer to the GGIG user guide for any further details or use the inbuilt help system by pressing F1.

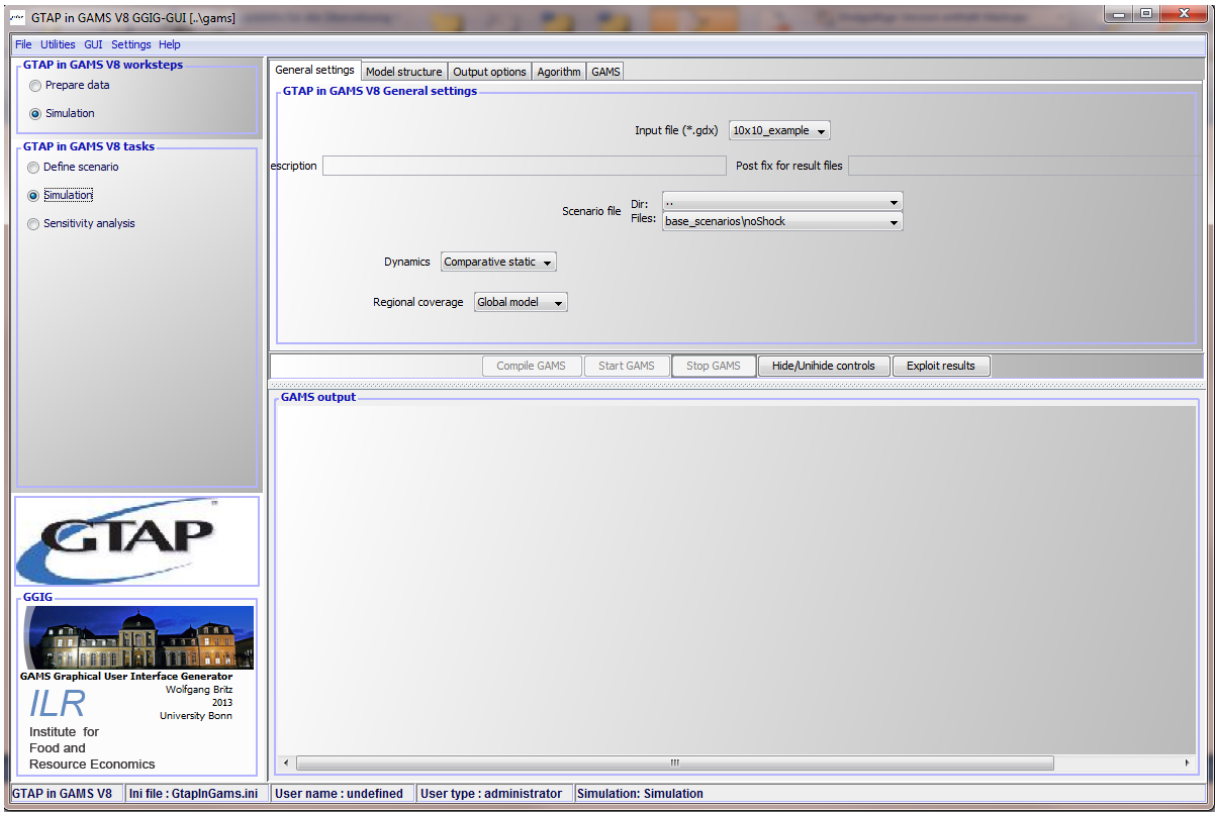
We would finally like to note that we did our best to test the combination of the GTAPinGAMS code and the GUI, but that improvements are certainly possible.

First steps

The installation comprises a set of results for testing the interface without the need of actually running GAMS code. You can now already look at these results by selecting on the left hand side the work step "Simulation":



You will see the interface for the "Scenario editor" which is discussed later. We skip that functionality here and move directly to the task "Simulation":



and next pressing the “Exploit results” button. The layout will change as seen below. Please click now once in the “Dataset” selection (otherwise you will get a warning message later):

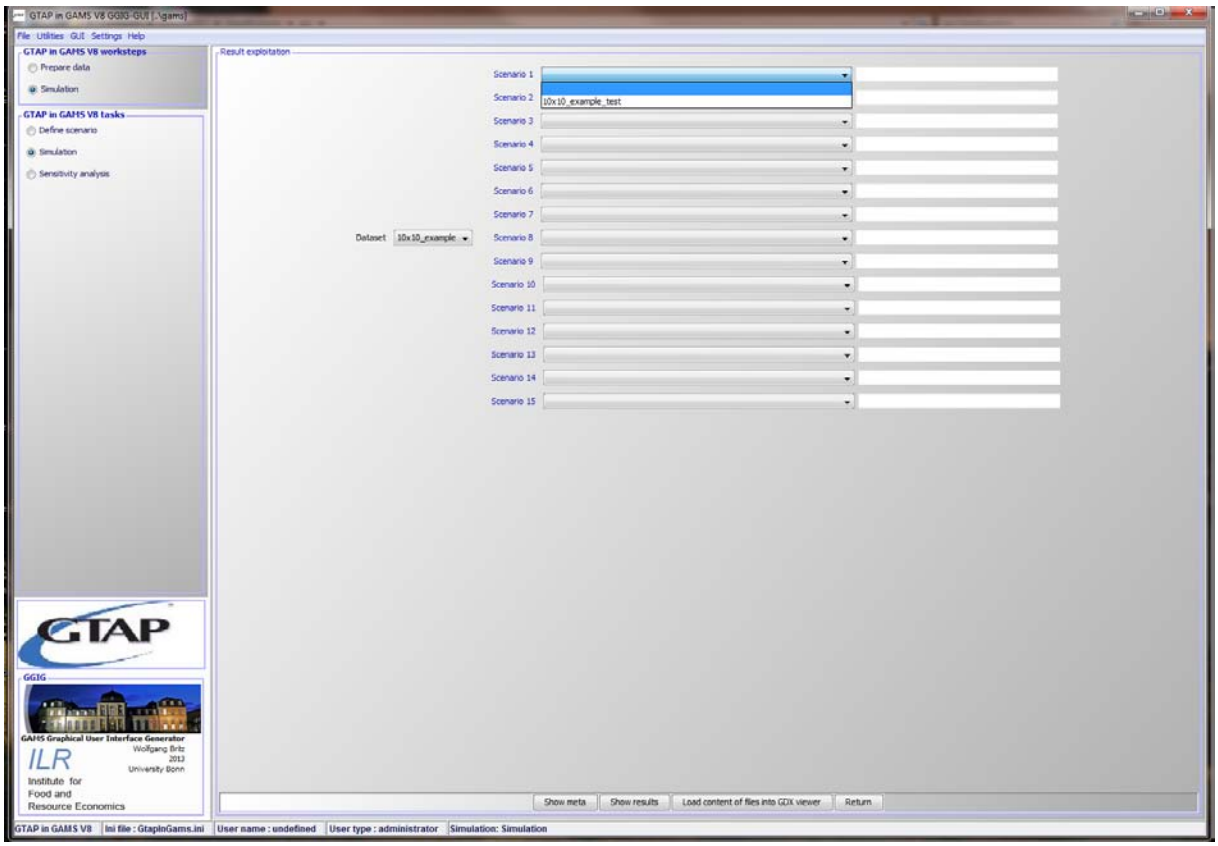
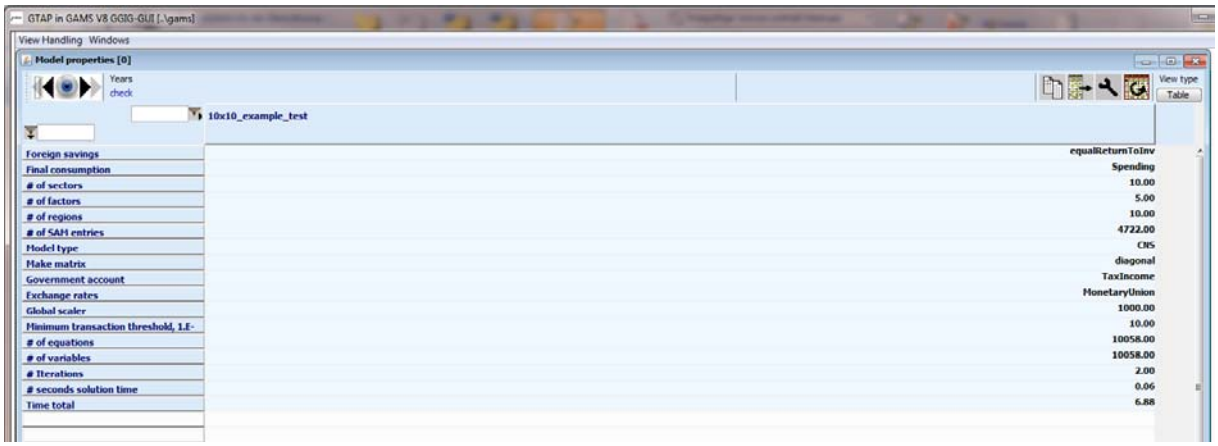


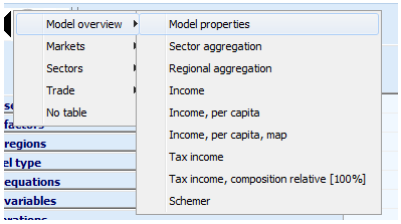
Figure: Scenario selection for exploitation

Select the scenario “10x_10_example_test” and press “show results”, you should see a table similar as the one shown below:



(Once you have produced your own results, you can find the data bases you used and the scenarios generated in the drop-down boxes, and you might combine different scenarios for comparison.)

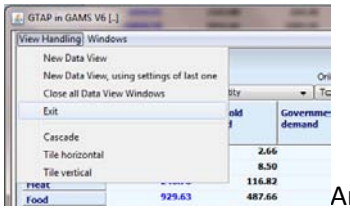
The “ ” view button in the upper left corner lets you select different views on your results:



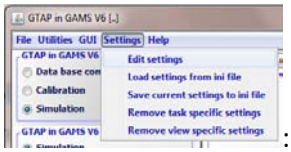
How to produce your own results is discussed in the following. Information on the different views is given in a section below.

Making the interface working with GAMS

In order to use the GUI to generate data sets or for simulation runs, you have first to register a GAMS executable. In order to do so, first leave the result viewer as shown below

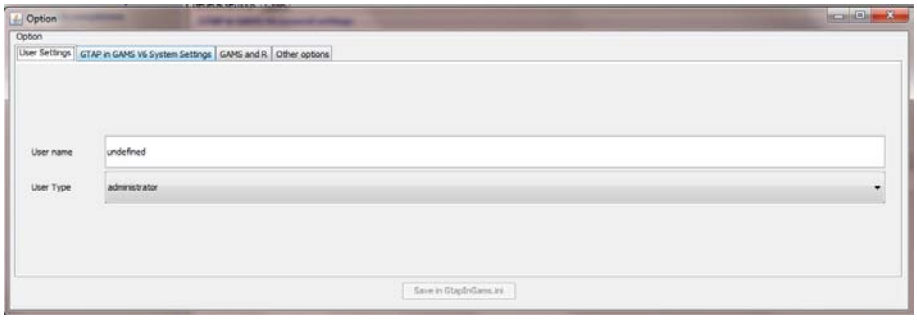


And open the setting dialogue from the menu bar

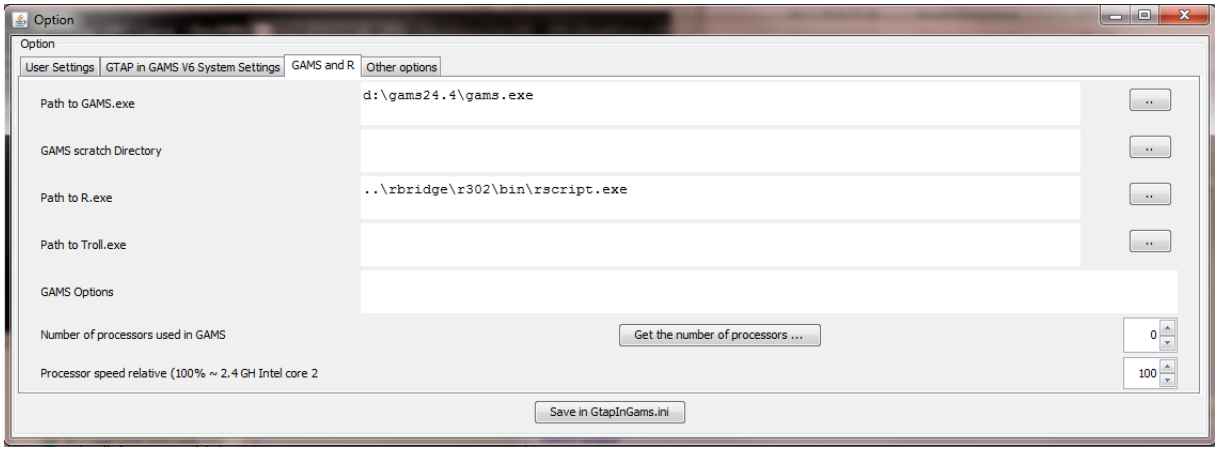


:

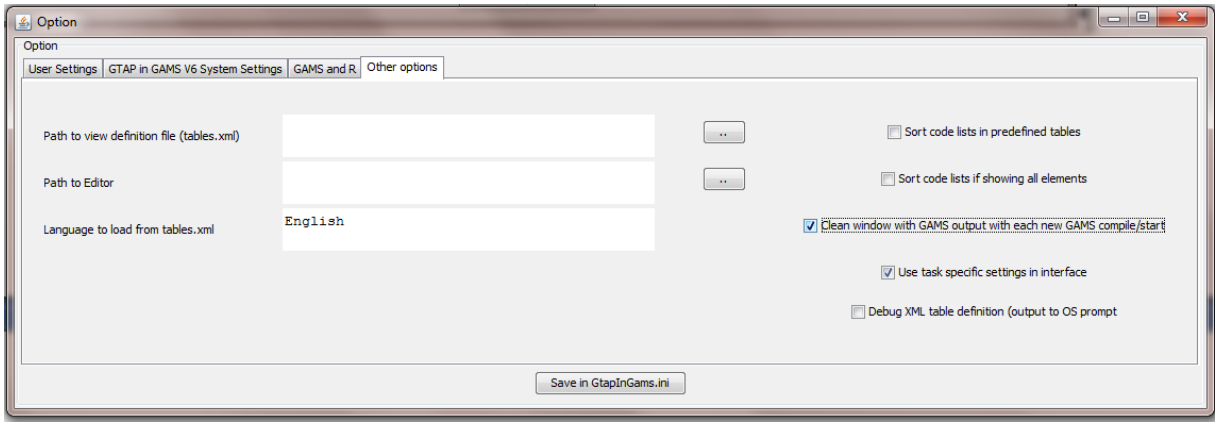
First, you should enter your name.



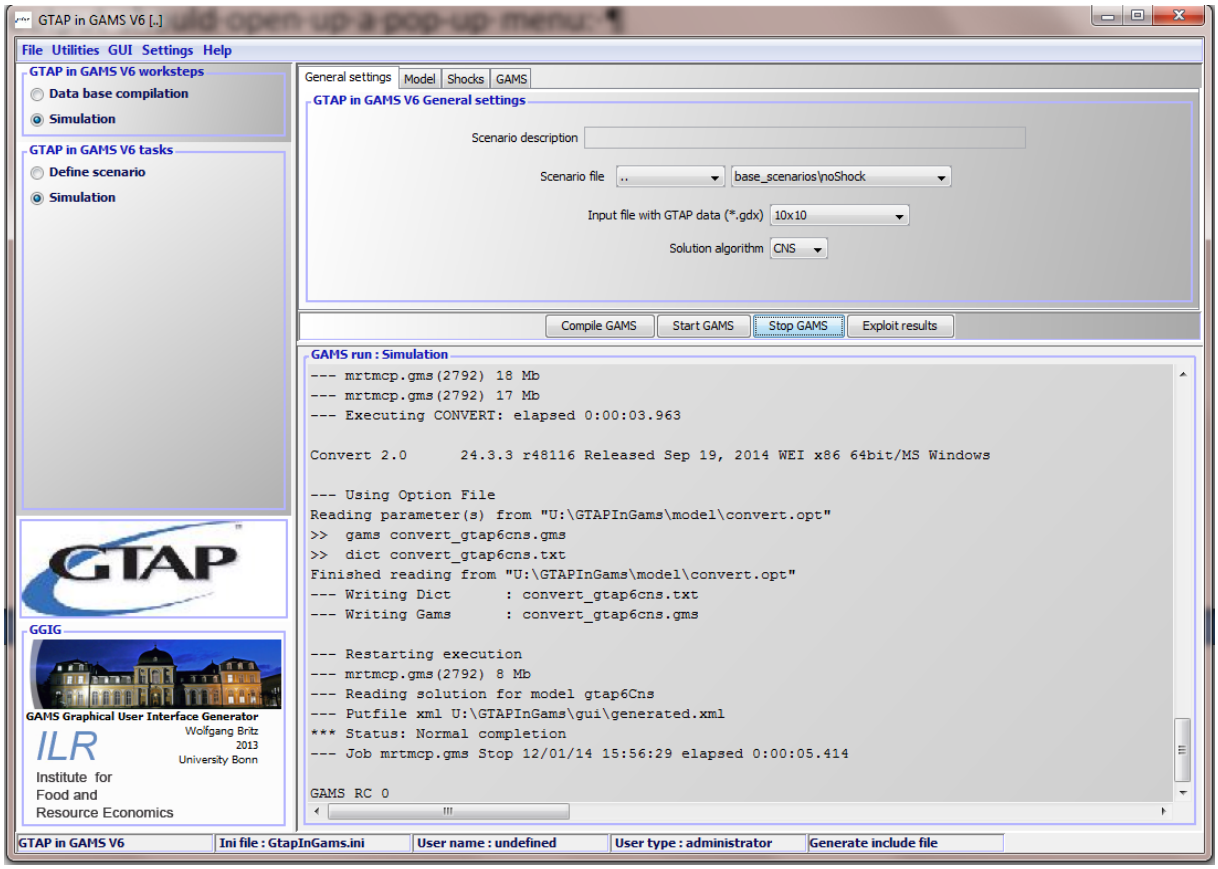
Next, click on the tab “GAMS and R” and either type in text field next to “Path to GAMS.EXE” which “GAMS.EXE” you would like to use. Alternatively, you can use the button to the right of the field to navigate to the directory where “GAMS.EXE” is found via a file selection dialogue. Please do not only enter a directory, but the full file name as shown below (and choose GAMS.EXE, not the user interface of GAMS, GAMSIDE.EXE).



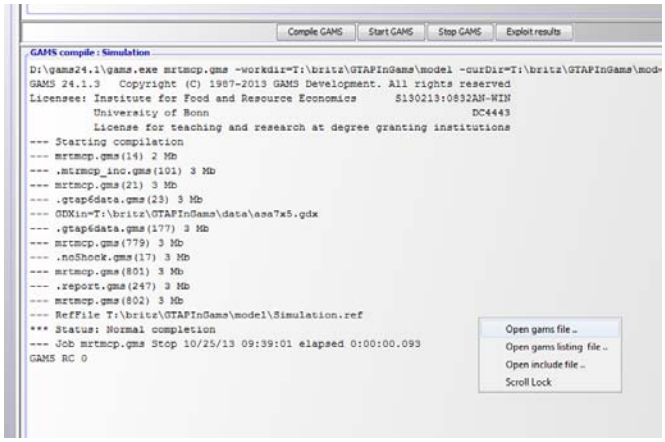
If you are using regularly a text editor, you can register it under “Other options”. You might also want to check the box “Clean window with GAMS output with each new GAMS compile/start”.


Option
Option
User Settings
GTAP in GAMS V6 System Settings
GAMS and R
Other options
Path to view definition file (tables.xml)
Path to Editor
Language to load from tables.xml
English
Sort code lists in predefined tables
Sort code lists if showing all elements
Clean window with GAMS output with each new GAMS compile/start
Use task specific settings in interface
Debug XML table definition (output to OS prompt
Save in GtapInGams.ini

In order to check if it worked, press the “Compile GAMS” button. You should see now the compile output from GAMS in the window below the button.



If you have registered an editor, you can also check its proper functioning: a right mouse click in the window with the output should open up a pop-up menu:



That pop-up menu offers you the options to open in your favorite editor the GAMS file, the listing produced by GAMS and the include file generated by the interface.

Notes:

- The use of the GTAP-AGR or GTAP-E module requires a data base with a full dis-aggregation of primary agriculture respectively detail for energy sector – switching on the module using a data base with insufficient sectoral detail will lead to compilation errors.
- You cannot use the global bank mechanism (Global equal returns to capital) if capital is declared immobile.

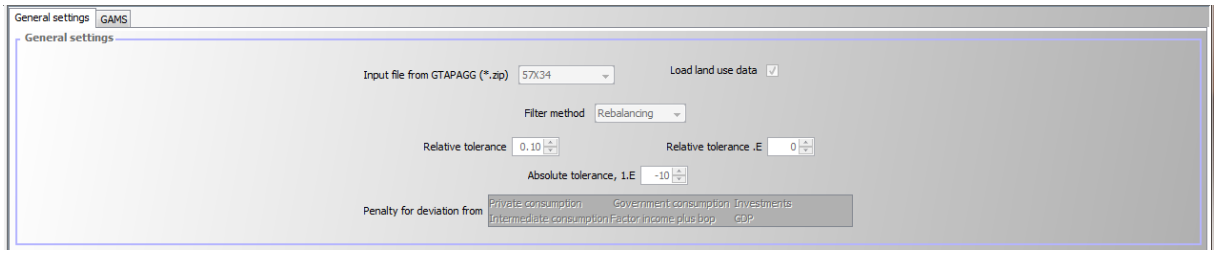
- A license either for PATH or CONOPT is required to solve the model. Using the pre-solve algorithm requires CONOPT, as does using the filtering approach when generating a data base.

Data base generation

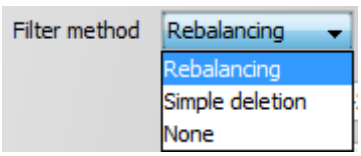
The GTAPAgg utility, which you should have received together with your GTAP data base, allows you to build a GTAP data base with a sector and regional aggregation chosen by you (see e.g. https://www.gtap.agecon.purdue.edu/products/packages.asp, a training video on you tube https://www.youtube.com/watch?v=QDBR0KqNuzE&feature=youtu.be or the first pages of http://economia.unipv.it/pagp/pagine_personali/msassi/QPA/materialQPA/Introduction%20to%20GTAP.pdf). That data base comprises different data sets stored in HAR (header array) files, a proprietary data format for GEMPACK. Mark Horridge has developed a program called HAR2GDX which converts a HAR file into the proprietary data format GDX used by GAMS. HAR2GDX is part of a GAMS installation and used by us to load output from GTAPAgg.

If you generated a data base with GTAPAgg intended for use with GTAPinGAMS, please copy the zip file generated by GTAPAgg to the "*data*" directory (or store the file directly there from GTAPAgg). In order to make the data available to the model, choose the workstep "Prepare data" and the task "Import from GTAPAGG". The file you copied in the data directory should be available in the dropbox under "Input file from GTAPAGG (*.zip") and selected by you.

If you want to use the GTAP-AEZ extensions, copy the "2007Luv81.tzip" in the data directory and check "Load land use data". Please note that once the land use data for the new data base version 9 is released, the filename is subject to change.



During data base loading, you can use a facility to additionally filter out small values, three options are available:



Choosing the option "None" will simply load the data base as it is, i.e., without filtering, which is the way the GEMPACK version operates. The following short section discusses the two other options.

Simple deletion

With simple deletion, transactions loaded from the HAR file are removed from the data base if they in absolute terms are below the threshold entered on the interface under "Absolute tolerance". With a value of 1.E-10, that deletion step is skipped. Afterwards, the transactions are formatted into a

SAM structure. The resulting SAM is then cleansed with the chosen absolution tolerance. No attempt is made with "simple deletion" to maintain the resulting SAM balanced. That option is mostly maintained as a fall back, in case the more refined rebalancing step normally recommended and discussed next should not work.

Rebalancing

The "Rebalancing" option uses more advances tactics to select transactions to delete, and perhaps more importantly, rebalances the resulting SAMs. As with simple deletion, first transactions loaded from the HAR file which in absolute term are below the chosen absolute threshold are removed from the data base. With a value of 1.E-10, that preliminary deletion step is skipped. It is generally not recommended to use absolute deletion thresholds above 1.E-6 in combination with rebalancing as the subsequent relative thresholds will anyhow apply more refined rules. Please note that rebalancing was only tested with CONOPT and that alternative NLP solver such as PATHNLP might not work satisfactory.

The consecutively applied filtering and rebalancing approach is an extension of the method and code developed by Tom Rutherford for "GTAPinGAMS" (see e.g. http://www.mpsge.org/gtap6/). It deletes component of the SAM depending on their shares on specific totals, according to the "Relative tolerance" entered on the interface:

- Domestic and imported intermediate demand of a commodity are dropped relative to its total output
- Private / government / investment domestic respectively import demand of a commodity are dropped relative to total private / government / investment domestic respectively import demand
- Trade flows of a product are dropped if both shares on total exports of that product and its exporter and on imports of that product and its importer are below the relative threshold
- Production is dropped if net production of a commodity, i.e. after intermediate use of that commodity in its own production is deducted, is below the relative threshold with regard to total net production

The filtering process imposes restrictions which should maintain the regional SAMs balanced. Additional constraints ensure that production activities require added value and intermediate inputs, if not already otherwise found in the data base.

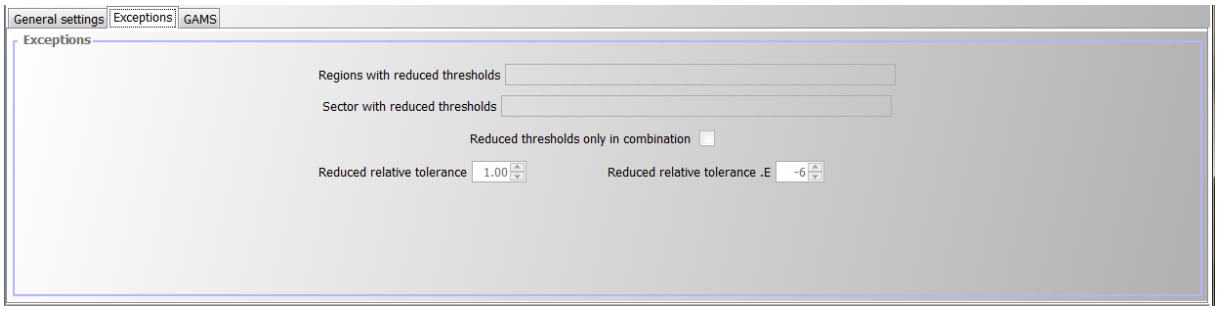
As filtering systematically removes elements from the SAM and the trade matrices, the process implies without further corrections shrinking the economies. During rebalancing, the algorithm can therefore add penalties for deviations from the following aggregate transactions:

By adding these penalties terms, the non-deleted entries (and thus most important transactions) tend to be scaled upwards. It is generally recommended to use these penalties terms. The code will also scale all non-deleted trade flows to approximately maintain the total volume of international trade.

The thresholds are stepwise enforced, starting with 10% of the desired final ones. Once the final thresholds are active, filtering is still applied several times until no small values are found any longer. The code should ensure that the resulting transactions are still fully consistent with each other, i.e. both the resulting trade matrices and the SAMs are balanced. The changes imposed by filtering and subsequent balancing are stored on the "itrlog" symbol in the GDX container with the final results. Inspecting how the stepwise enforcement of the thresholds impacts on the number of non-zero items can inform on an appropriate level for tolerances to be used.

Thanks to balancing, also rather dis-aggregated versions of the model with large number of sectors and regions can be used. The biggest impact of the filtering is typically on transactions related to bi-lateral trade flows. Here, often 50% or more of the flows account for only 1% of the total value of these transactions. Thus, tiny changes in the relative tolerance can have a considerable impact on the number of deleted transaction, and one might need to experiment with settings in the range around 1.E-1 to 1.E-4 to find a compromise between sparsity and the implied changes on structure of the economy. For very large data sets (e.g. a 1:1 version) filtering thresholds above 1% might be needed to yield reasonable model sizes. The user can additionally define a minimum number of transactions to be kept, which reduces the need to experiment with different thresholds as the filtering process will stop once less than the desired number of transactions is reached. Tests with the model have shown that models with more than 400.000 transactions cannot always be solved, especially if the global bank mechanism active. A close look at the filtering statistics is recommended, to avoid sharp impacts on the structure of the economy.

Special treatment for specific regions and sectors



When building a data base for a project, it might be desirable to apply less aggressive filtering thresholds for specific regions and / or sectors in the focus of the application. The algorithm therefore allows defining lists of regions / sectors with accompanying specific thresholds. The codes for regions / sectors needs to be inputted in the two text fields. "Reduced thresholds only in combination" will apply the different threshold only to the intersection of the inputted regions and sectors, otherwise, all regions and sectors inputted will be receive different thresholds. Take an example where you enter for regions "xoc" and for sectors "pd". If "Reduced thresholds only in combination" is NOT switched on, all transactions of the region "xoc" and all transactions for the sector "pd" will be treated differently. If the "Reduced thresholds only in combination" is active, only the transaction relating both to "pd" and the region "xoc" are exemptions.

However, filtering for the remaining sectors / regions has still an impact on these exemptions. For example, if production of a sector in a region is dropped, the related export flows need to be dropped as well, affecting potentially transactions in regions and sectors where tighter thresholds are used. Tests have however indicated that very few transactions are lost in regions/sector where stringent thresholds are applied as long as the overall filtering thresholds are not too aggressive.

Model size and solution behavior

The model template including the GTAP-AEZ and GTAP-AGR modules was intensively tested under a suite of test shocks, applied simultaneously to all regions and sectors: 50 % reductions in tax rates (direct, consumption, factor, tariffs/export subsidies), 20% endowment changes and 10 % tfp shocks for all primary factors. The model was tested in three configurations: (1) the GTAP standard model with fixed allocation of foreign savings and (2) alternatively the global bank mechanism as a default in the standard GTAP model, and (3) the global bank mechanism in combination with GTAP-AGR and GTAP-AEZ. The tests used a full sector dis-aggregation (57 sectors) and varyingly sized regional aggregations (10,24,36,45,56,68). Additionally, a 10x10 model was solved for which the GTAP-AGR and GTAP-AEZ modules cannot be used due to missing sectoral detail. The parameterization was kept at defaults. Most of the tested models are large with regard to the number of sectors and countries compared to applications of GTAP reported in publications. All tests were also run solving the model as MCP in PATH and as a CNS in CONOPT.

Table 2: Data bases used in the standard tests

Model size	Filtering thresholds (relative tolerance, minimum # of transactions)	Resulting non-zeros in global SAM, including trade flows	Model size (GTAP Standard, maximal number of variables substituted out)
10x10	None	~4.700	~8.000
57x10	None	~35.000	~92.000
57x24	None	~157.000	~225.000
57x36	1%, 160.000	~120.000	~96.000
57x45	1%, 160.000	~130.000	~99.000
57x56	1%, 160.000	~133.000	~98.000
57x68	1%, 160.000	~135.000	~99.000

Note: The number of transactions accounted for during filtering is somewhat higher that the resulting non-zero SAM entries

GAMS version 25.5.4 was used, on a computer server, in combination with a beta-release of CONOPT4 which executes certain part of the NLP algorithm in parallel. We would like to acknowledge the continued support of Arne Drud, especially to let use CONOPT4 for the tests. Results with CONOPT3 are somewhat slower. The pre-solve algorithm with five pre-solve steps was applied. The aim of these tests was not only to ensure a stable numerical implementation of the model, but also to gain experiences with solution behavior on larger dis-aggregations.[3] The times reported in the table below are for a full model run without post-model processing switch on, but including the time needed to store all symbols in a GDX container, i.e. they cover loading the data base, model calibration, a benchmark solve and solving the shock.[4] The test show the expected more than linear increases in solution time if model size and complexity expands. Solution times with a fast multi-core laptop will be about double the times reported below.

[3] The aggregation definition files (*.agg) used are found in the repository in the data directory. The test shocks can be found in the "scen" directory.

[4] The distribution which includes the GUI also comprises the described above test suite for the model, realized as an input file for the batch utility (gui\testbatch.txt). It is recommended to run the test suite after changes to the model code.

Many of the test solves will actually run even somewhat faster without the pre-solves switched on. However, if the solver fails to solve the shock without any pre-solves quickly, it can often spent several minutes until all infeasibilities are removed. As the potential further gains in solution speed by switching off the pre-solves are limited, we opted to show in here results obtained with the default settings. For small models and special applications such as sensitivity analysis, it might however pay off to check if the overhead of using the pre-solves can be avoided.

Table 2: Solution time with different data bases and model configurations, CNS

Data base	GTAP standard, Fixed allocation of global savings	GTAP Standard, Global bank mechanism	GTAP Standard + GTAP-AGR+GTAP-AEZ Global bank mechanism
10x10	10 sec	10 sec	(not possible)
57x10	30 sec	30 sec	35 – 45 sec
57x24	1. min 30 sec	1 min 30 sec – 2 min	1 min 40 sec – 2 min 30 sec
57x36	1min	1 min – 1 min 30 sec	1 min – 1min 30 sec
57x45	1 min 10 sec	1 min 10 sec – 2 min	1 min 15 sec – 3 min
57x56	1.min 20 sec	1 min 20 sec – 3 min 20 sec	1 min 20 sec – 3 min 20 sec
57x68	1 min 30 sec	1 min 30 sec – 3 min	1 min 30 sec – 11 min[5]

Even if additional tests showed that rather large models such as a 57x82 variant can be solved even without any filtering at least on some larger shocks in around ten minutes, such tests failed with a model using the full resolution of the data base (57x140), but also with certain larger shocks on the 57x82 variant in any reasonable time. The tests were only possible with the beta version of CONOPT4, as CONOPT3 will exceeds its internal memory limit of 8GB on very large models (around 600.000 equations which implies a SAM with about the same number of entries). That implies that users might run into very long solution times under a combination of larger shocks and very detailed data bases or might even not be able to load the model into the solver.

Analysis of changes in simulated welfare changes under a multi-lateral trade liberalization scenario which dismantled 50% of all import tariffs and export subsidies indicate that filtering can influence substantially results obtained. Unfortunately, the same holds for changing the regional and sectoral resolution of the model. We can therefore generally only recommend running highly dis-aggregated models with limited filtering. For further detail on solution times and aggregation bias due to sectoral and regional aggregation or the use of filtering, refer to Britz and VDM 2015 (ref) who provide a larger sensitivity analysis with the standard GTAP model in GAMS.

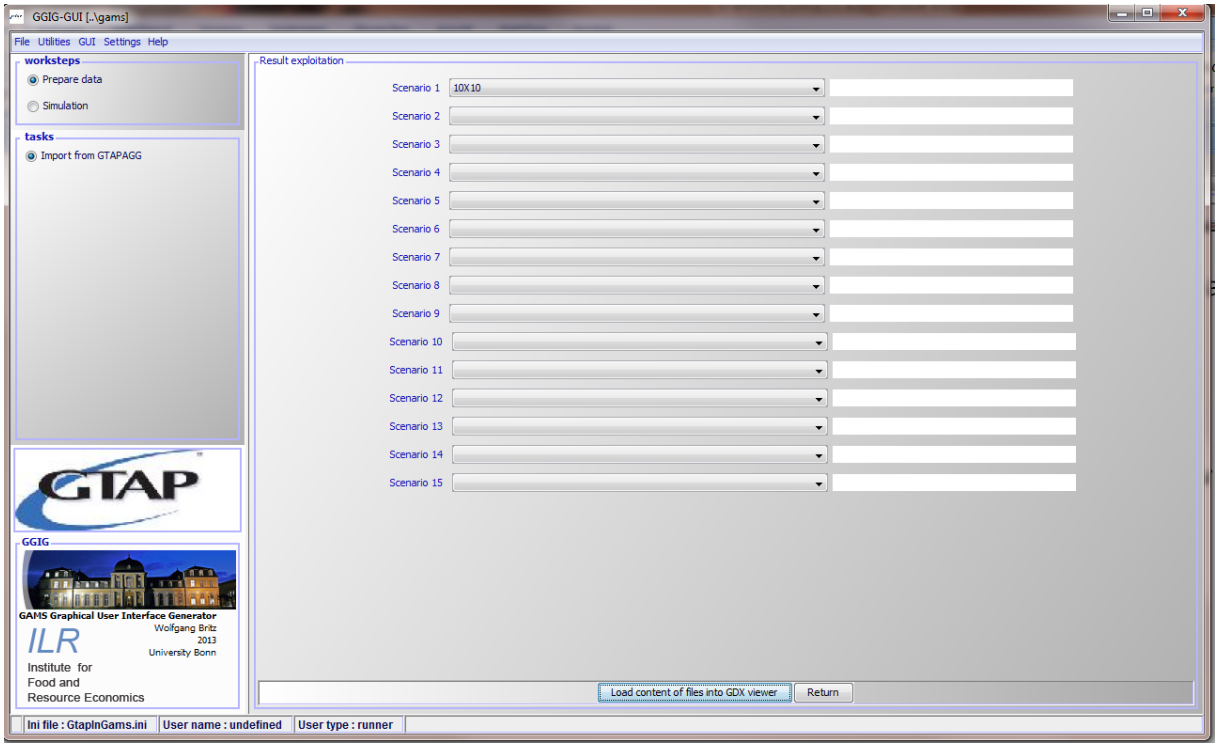
[5] The long solution times occurred under the direct tax cuts. As the standard GTAP model assumes that FDI is taxed at the same rate as domestic one, the direct tax cut impacts fully the expected returns to capital by foreign investors which drive the distribution of foreign savings under the global bank mechanism. A 50% reduction as simulated in the test shock can provoke very large differences in expected returns depending on the regional resolution of the data base. Thus, a new equilibrium can require massive changes in investments to decrease or increase the regional capital stock in order to drive marginal returns to capital up and down towards the global average. For some regions, that scenario might not feasible without fixing some variables at their bounds. Solving the model as a MCP should yield automatically that solution. However, solving as a CNS will yield infeasibilities in that case. Therefore, an algorithm is comprised in the GAMS code which tries to find, based on equations becoming infeasibility, which variables to fixed, which however requires to solve the model potentially several times. That algorithm can also fix similar problems which might not be related to the global bank mechanism.

The tests have shown that in combination with the pre-solves, solving the model as a CNS with CONOPT is usually faster if the model is larger compared of using PATH and solving as a MCP. Very small models are typically faster solved with PATH. Having both solvers available for tests might hence pay off if solving large models is part of a project. And clearly, certain policy instruments such as production quotas are best captured by a KTT-condition embedded in a MCP model.

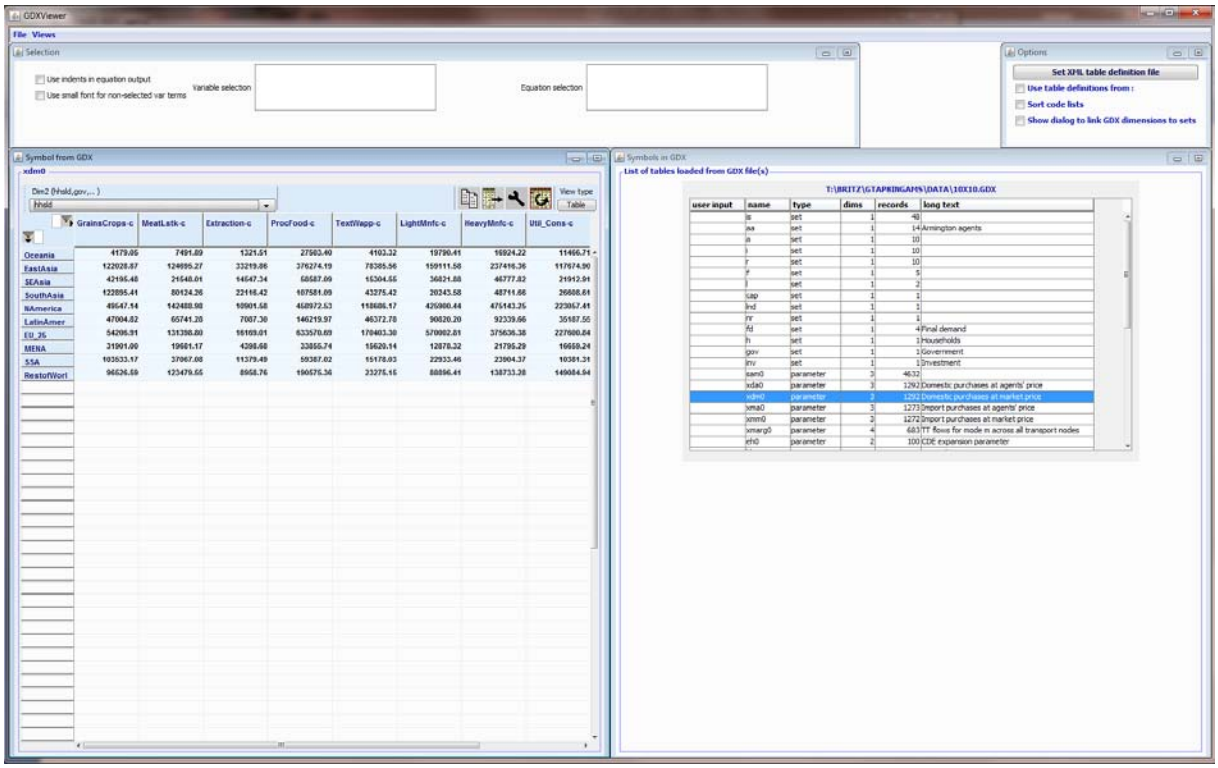
Checks for medium sized models have shown no differences in the results produced by CONOPT or PATH. That seems to indicate that once a model is declared feasible, the results can be used. Care should be clearly given when using PATH as a MCP solver after structural changes to model code: MCP solvers might e.g. accept a fixed variable paired to an equation which results in a non-square system, whereas solution as a CNS will throw an error in that case.

Inspecting the resulting data base

You can check if loading and filtering worked by pressing on “Exploit results”, selected as shown as output file generated, a GDX comprising different parameter and set definitions:

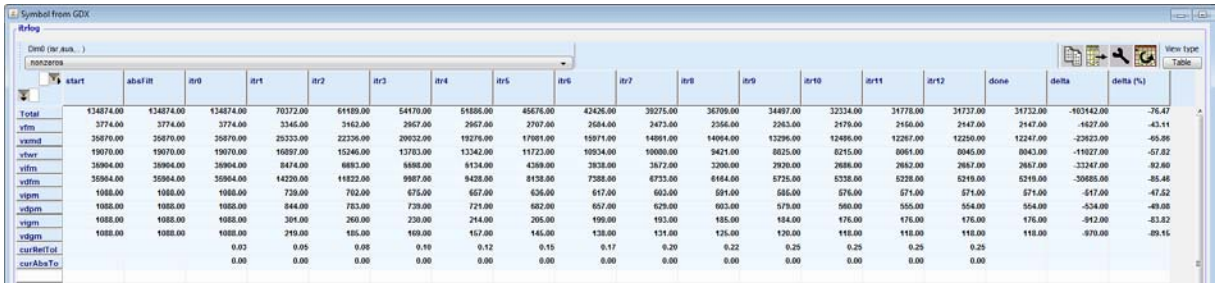


Once you press “Load content of files into GDX viewer”, you can inspect the individual symbols used in the model:



Double clicking on any on the symbols in the table on the RHS will open a new window as shown above.

The effect of using filtering and the rebalancing can be checked with the symbol “itrlog” which shows totals ,aggregated over commodities or sectors, for each rebalancing step (per country and “trace” at global level), the number of nonzero at global level and related changes. The example below shows the output from filtering a 32x34 case. Filtering removes about 76% of the original almost 135.000 non-zero items (see row “Total”). The line “curRelTol” shows the applied threshold in the iteration, which is increased stepwise to the desired maximum of 0.25%. Inspecting the relation between the dropped items and the cutoff used in that preliminary iteration can help to find a good compromise between model size and a too aggressive filtering where also more important transaction are wiped out.



Pre- or post-model aggregation

The standard case in GTAP applications is a project specific pre-model aggregation. Given the total size of the data base – there are already about 1 Million non-zero trade flows reported – pre-model aggregation cannot be avoided, it yields smaller models which are faster to solve, easier to debug and typically also numerically more stable with less small values being present. However, aggregation comes clearly at a price: not only might detail of interest for the application be lost, but also peaks of

policy instruments etc. wiped out. That can have important effects, Ko and Britz 2013 show in example applications that one might increase simulated welfare from FTAs by simply increasing sectoral and regional detail.

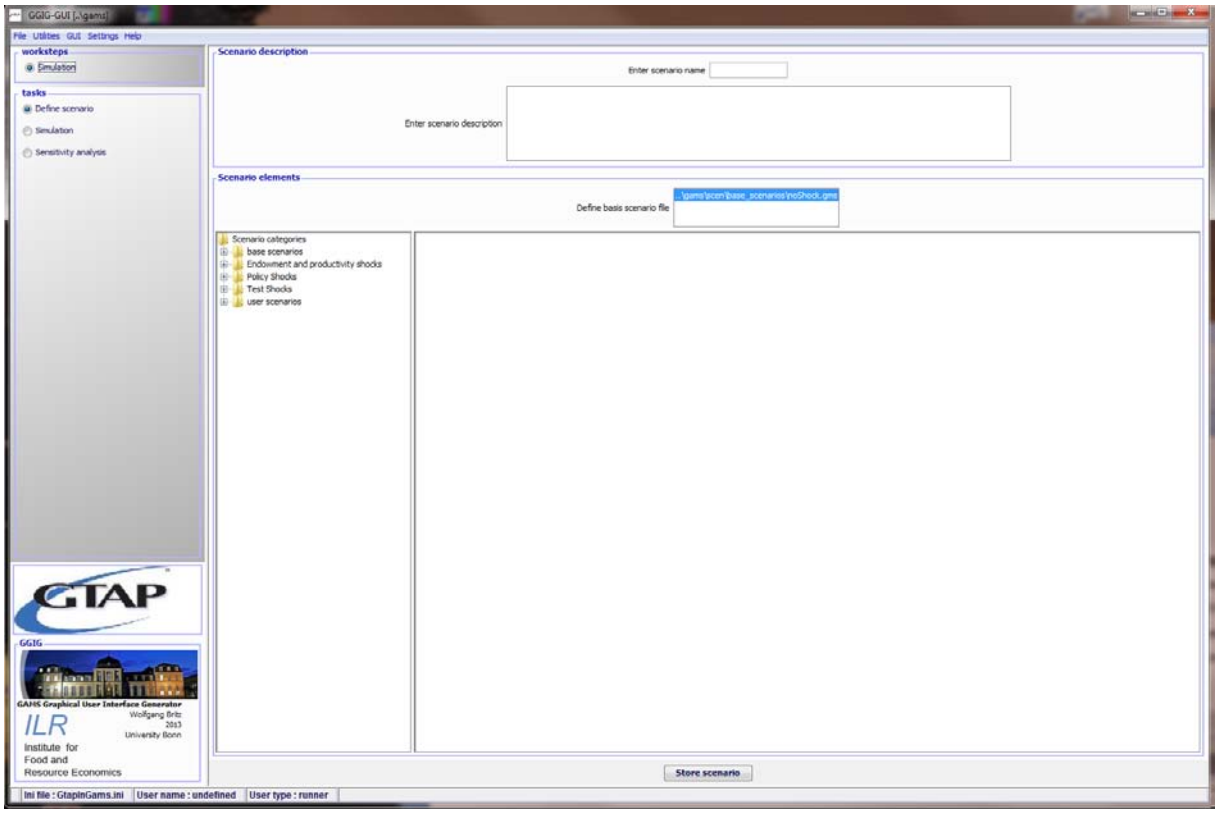
GTAPinGAMS tries to soften the decisions on what regional and sectoral aggregation level to use based on three features. Firstly, filtering and rebalancing of a data base can help to solve models with rather high number of sectors and regions (but clearly, the filtering process will also introduce aggregation bias). Secondly, the equations of the model had been carefully arranged and scales and bounds for variables introduced to stabilize the solution behavior of the solvers and to speed up solution time. And thirdly, post-model processing allows for a second aggregation step for reporting purposes. To do so, the user has to provide a second aggregation definition file (*.agg, can be produced by GTAPagg). See also Britz and VDM 2015 for discussion of solution times and aggregation bias.

Scenario runs

Defining the scenario

The interface helps with the definition of scenarios – it has a set of example files with shocks which can be combined and edited directly in the GUI. Alternatively, you can use any text editor and define a shock in GAMS and store the file somewhere under “\gams\scen”.

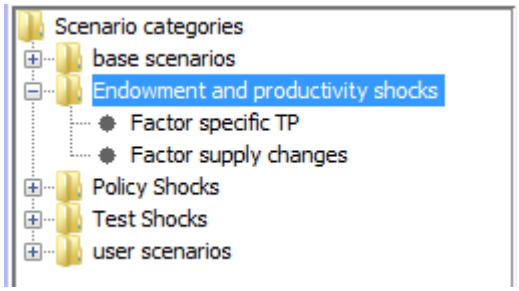
In order to use the in-built scenario editor, click on “Define scenario” and the interface should look like below.



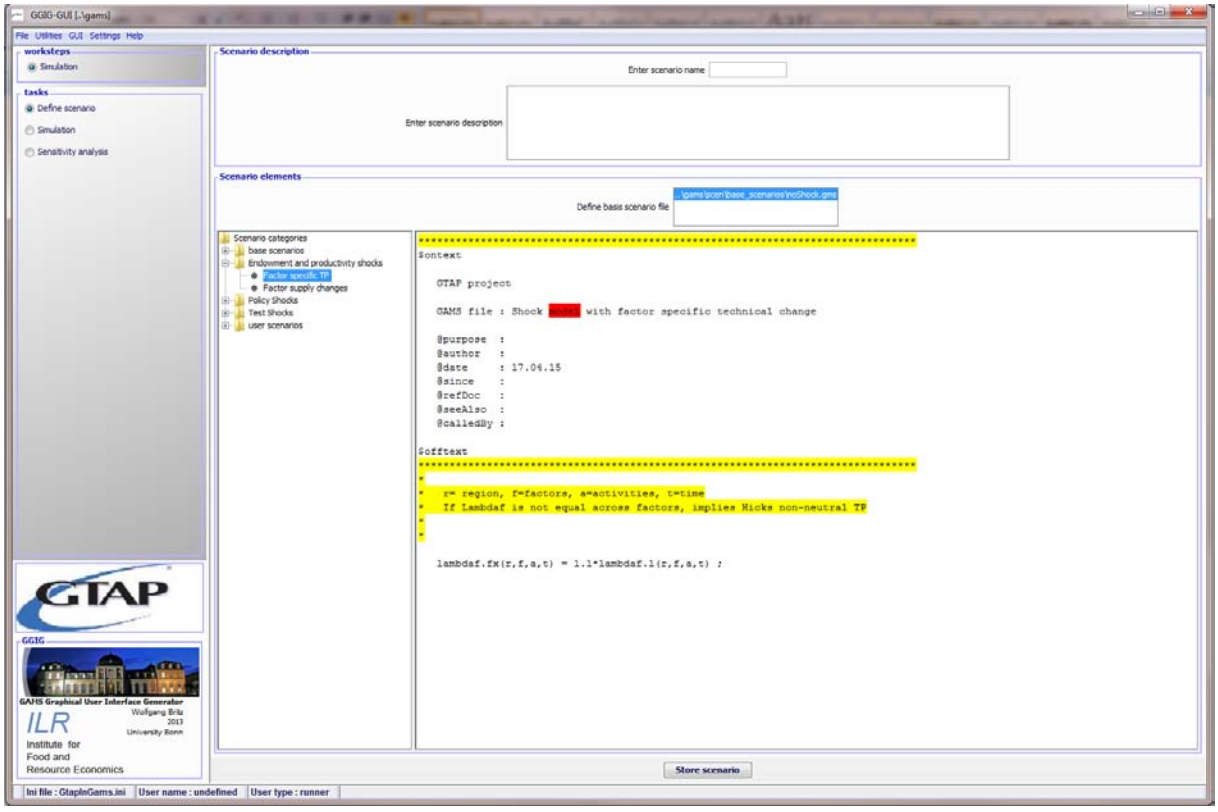
Currently, there are two one base scenarios:

- **No shock** – that is simply an empty file
- **RecDyn** – a test implementation for a recursive-dynamic baseline.

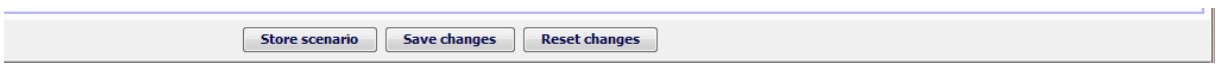
- In order to add the content of pre-defined shock files to your base scenarios, click on a tree element, e.g. “Endowment and productivity shocks”. You should be able to see a list of files as below:



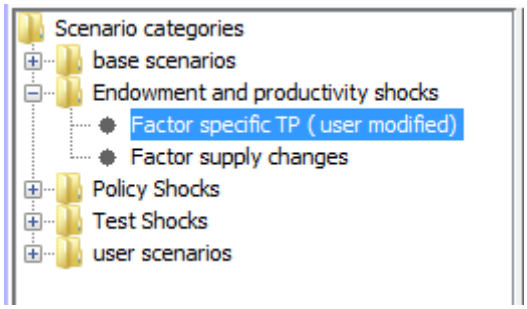
Double-click for example on “Factor specific TP”, and the content of that file is loaded in the editor on the right as seen below:



You can now modify the content with the editor by typing directly in the right hand window. Once you shave edited the code, additional buttons will appear:

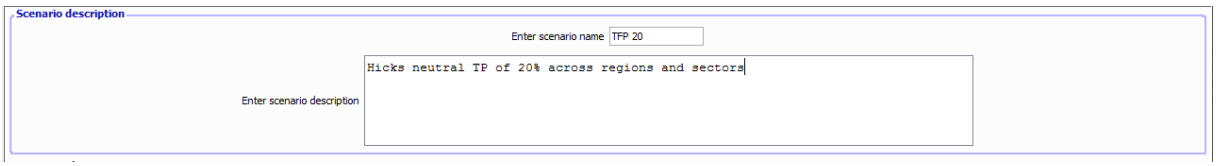


The list with the scenario groups and shock files will now indicate that you have made changes to the shock file.

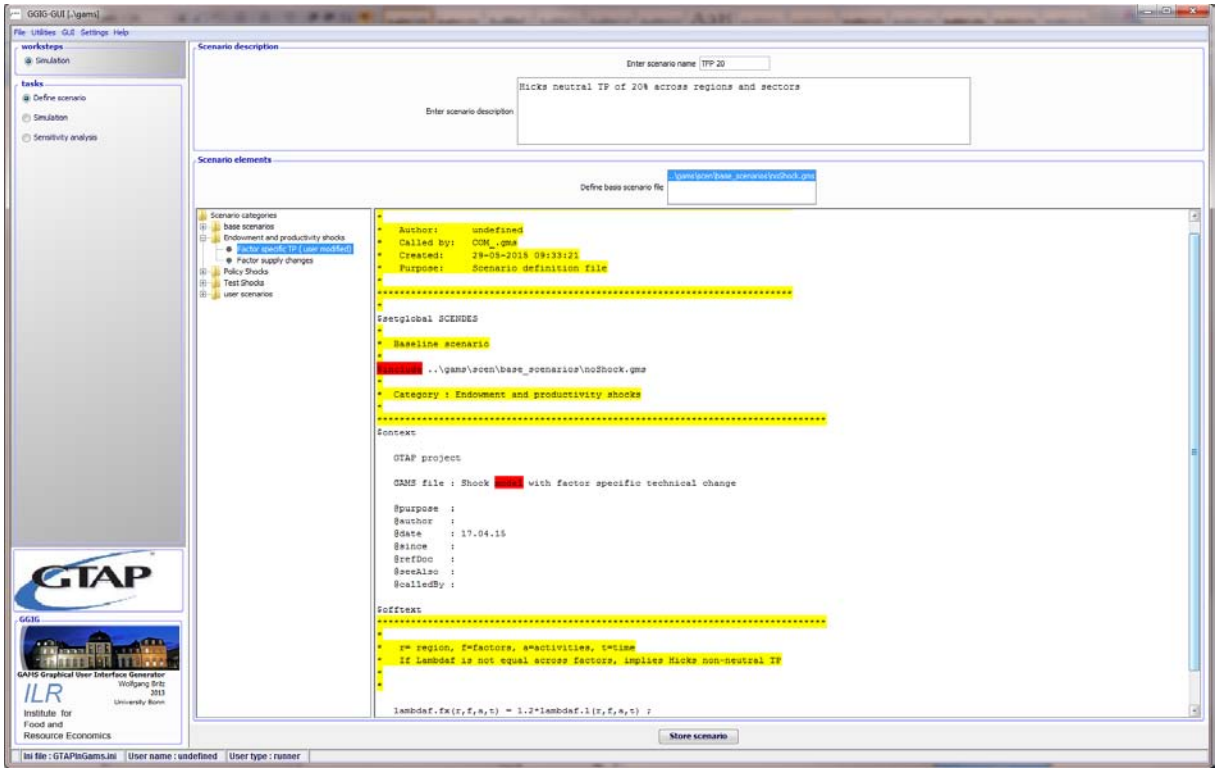


In order to save your edits, press “Save changes”. If you do not press “Save changes”, your edits won’t make it in final shock file generated. You can now double click on further files to add them to your shock file. The files you have currently combined are shown in blue; you can also deselect some of these files again.

Once you have chosen the files you want to combine and potentially edited (and saved!) your changes, you should now enter name for the shock file to create, e.g. TFP20 and a description of the scenario as shown below:



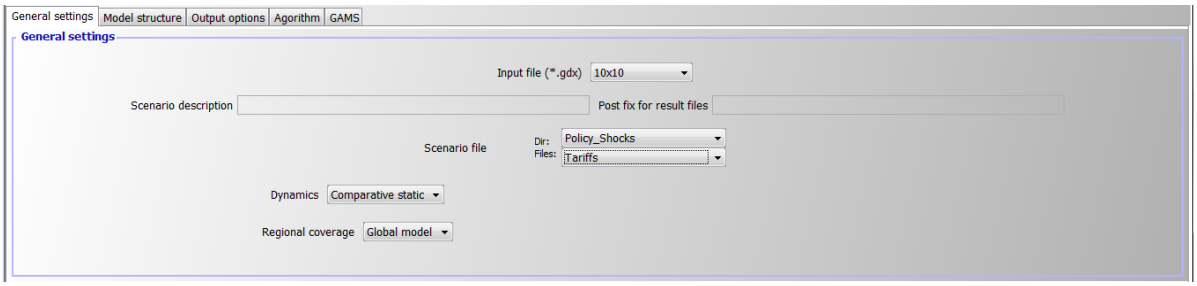
Afterwards, press “Store scenario”:



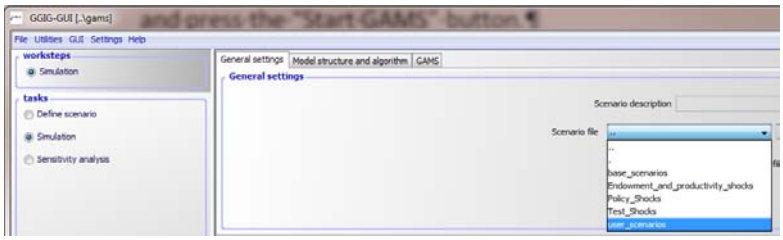
The window shows now the file as stored in the disk and included by the standard GTAP in GAMS model driver, when you use it to run a scenario.

Note: The scenario editor can also be used to edit parameters settings for the model. Several files which show the parameters used in different modules are provided.

Running the scenario



In order to run the scenario, click on “Simulation”, select first the directory with the user scenarios as shown below, and then pick the second box with scenario you just generated

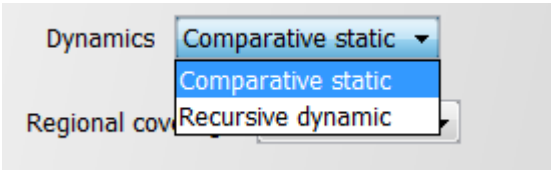


and press the “Start GAMS” button.

That should run your counterfactual against the reference and generate results in a GDX file.

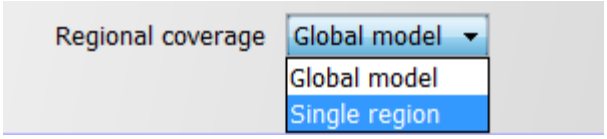
The tab “General settings” gives you the following additional input possibilities:

- Entering a *scenario description*: Normally, the output files generated are named of the scenario file used. That name can be overwritten with content of that text field.
- *Post fix for result fil*es: The post fix is appended to the names of the output files. It can e.g. used to differentiate versions using the same shock, e.g. results generated based on different closures or by using different modules.
- Choice between the *Comparative static or Recursive dynamic version* of the model:

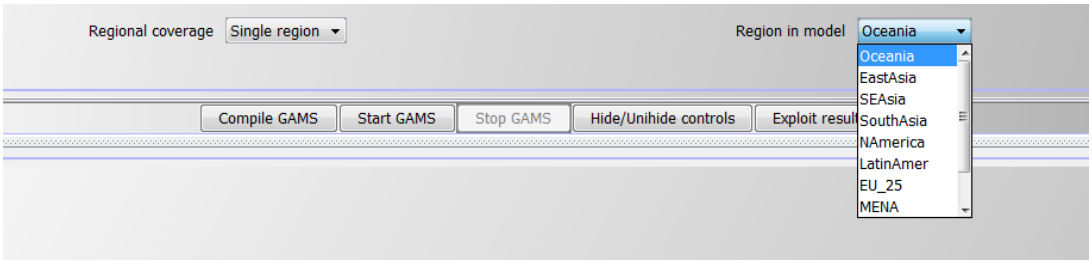


  Note that a simple example scenario which shows how to introduce shocks dynamically can be found under “scen\base_cenarios\regDyn.gms”. If the “Recursive dynamic” option is switched on, a drop down box is added which allows to define the number of years:

- Choice between the *Global Model or Single Region version* of the model:

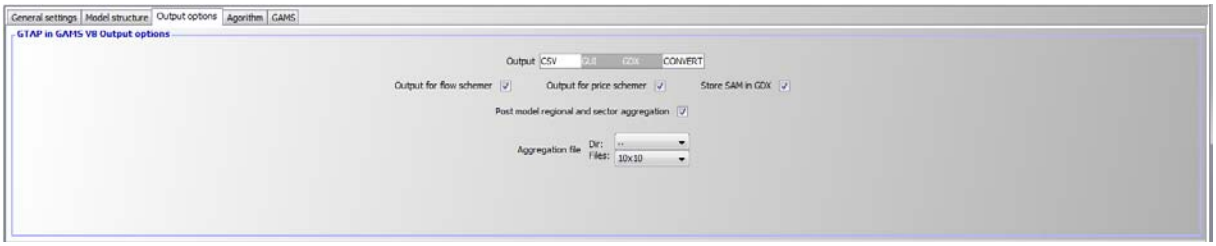


If the single region mode is switched on, a drop down box appears where the region can be chosen:



In single region mode, fob price of other regions are fixed and imports to other regions react to changes in the region's cif prices according to the share equations of the lower Armington nest at fixed total import quantities and aggregate import prices.

Output options



Also, make sure that output for the GUI is selected:

- CSV will generate a SAM and some other core results and store it in CSV format under "results\run"
- GUI will generate a GDX container with a parameter for use with the exploitation tools (see below)
- GDX will store all other GAMS symbols (parameters, variables, equations, sets etc.) in a GDX container, including output for the GUI if selected
- CONVERT will generate output for the "Equation and Variable viewer" which is similar to AnalyseGE (see section below).

If you are interested in a graphical depiction of flows and price changes (see chapter Overview on existing views for exploitation below), you can deselect these options to save some memory and speed up post-model processing.

The same holds for the post-model aggregation. The screenshot below shows what post model aggregation contributes: in most views, results are not only available in the regional and sector dis-aggregation of the underlying data base, but additionally also aggregated results as shown below.

Demand, by product [0]

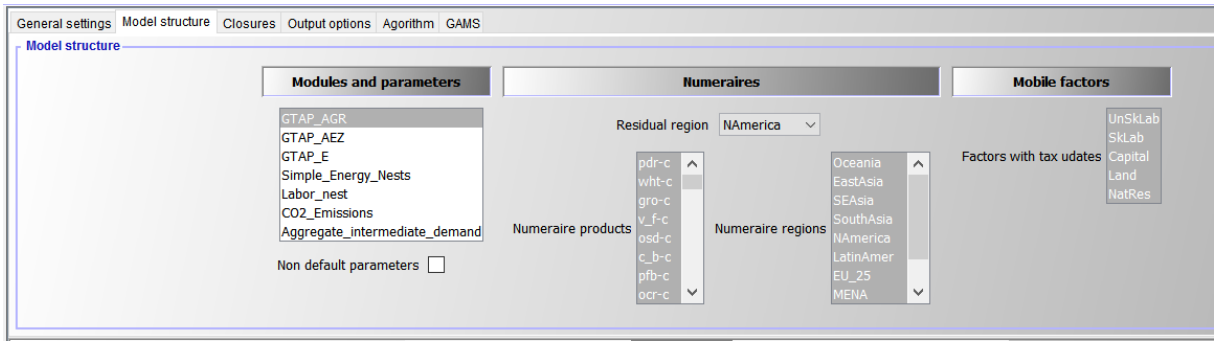
Regions: World | Items: Quantity | Origins: Total

TARIFFS

	Total	Household demand	Government demand	Intermediate demand	Investment demand	Export demand	International transport demand
Total	137582,97 0,41%	31212,10 0,30%	9534,91 0,02%	54631,16 0,01%	12416,49 -0,69%	23091,78 2,03%	591,84 4,30%
Agriculture	2058,32 0,43%	800,55 0,29%	1,87 12,82%	504,73 0,58%	0,57 -0,73%	413,50 0,78%	
Mining and Extraction	5474,23 2,91%	58,66 -1,99%		2484,97 0,57%	1,05 0,20%	1542,53 6,36%	
Light Manufacturing	17319,36 1,02%	3950,65 1,51%	18,43 1,91%	4854,85 0,74%	985,41 -0,50%	5116,43 1,23%	
Heavy Manufacturing	32869,18 0,68%	2062,65 0,66%	53,34 0,27%	15055,72 -0,24%	2527,85 -0,77%	11201,62 2,35%	
Utilities and Construction	11765,99 -0,44%	509,02 0,28%	9,54 -0,57%	3485,60 -0,59%	7415,18 -0,64%	292,50 5,26%	
Transport and Communication	22034,38 0,20%	9171,68 0,21%	67,26 0,88%	10126,40 0,09%	593,31 -0,63%	1725,47 1,05%	591,84 4,30%
Other Services	46061,50 0,01%	14658,88 -0,01%	9384,46 0,01%	18118,90 0,01%	893,11 -1,15%	2799,73 0,40%	
Total	137582,97 0,41%	31212,10 0,30%	9534,91 0,02%	54631,16 0,01%	12416,49 -0,69%	23091,78 2,03%	591,84 4,30%
Agriculture	2058,32 0,43%	800,55 0,29%	1,87 12,82%	504,73 0,58%	0,57 -0,73%	413,50 0,78%	
Paddy rice	21,36 -1,22%	6,28 -0,03%	0,19 -1,34%	11,79 -1,76%		1,55 -1,52%	
Wheat	99,08 -0,12%	12,88 0,28%		26,67 -0,05%		29,76 -0,24%	
Cereal grains nec	124,97 0,45%	48,10 0,76%	0,42 22,49%	17,66 -1,59%		34,18 0,75%	
Vegetables, fruit, nuts	910,70 0,70%	477,79 0,29%		252,22 1,14%		114,20 2,43%	
Oil seeds	133,49	8,28		58,67		33,27	

Model structure, parameterization and factor mobility

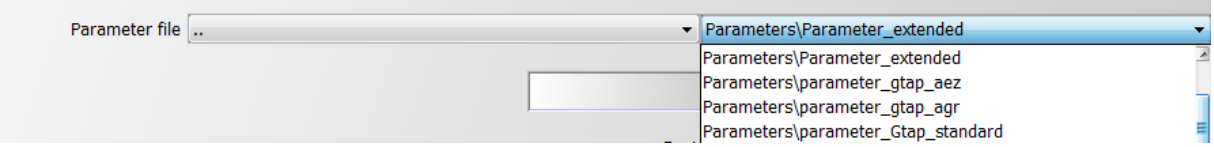
You might also select from a so far limited set of different model setups:



Modules and parameters

Currently, different modules and extensions as shown above can be added to the standard GTAP model. Please note that these modules are not a full replica of the GEMPACK implementations. The simple energy nests allow for substitution of primary and secondary intermediate demand for energy inputs. The labor nests – which is also part of GTAP-E – depicts substitution between skilled and unskilled labor. The "Aggregate intermediate demand" extensions reduced the number of Armington agents to four: final, government and investment demand and aggregate intermediate demand. The latter replaces the sector specific nests in the GTAP standard model.

Non-Default parameters can be used with the model. If “non-default parameters” is not switched on, the parameters of the standard GTAP model as provided by GTAPAgg are used. If the option is switched on, a file from disk can be chosen:

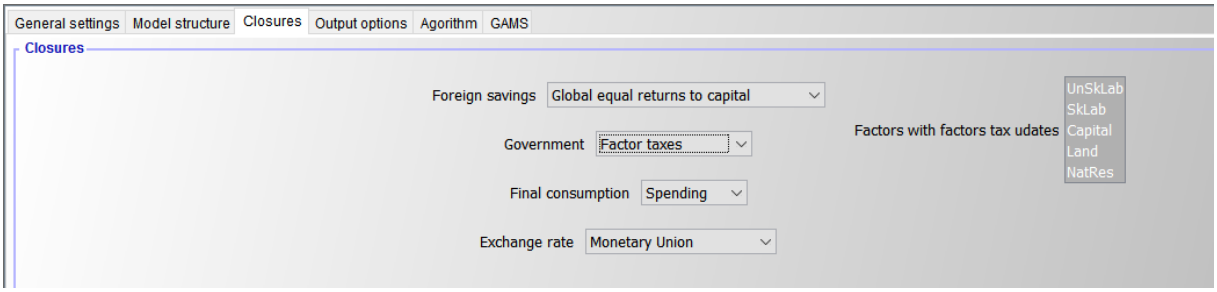


The possibility to use non-default parameters is important as certain model extensions are driven by parameter settings. To give an example: in the standard GTAP model, the transformation elasticity between domestic output and exports is infinite, such that producer prices between the two destinations are equal and physical balancing is used. The GAMS version allows alternatively using a CET structure where prices differs and consequently, a non-linear quantity aggregator is used. That feature can be switched on by providing a parameter file in GAMS format where a CET elasticity different from infinity is set.

Numeraires

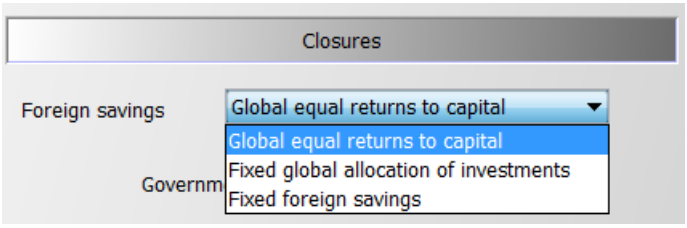
- The residual region closes global savings (see equations walraseq, savfeq and yieq in “model.gms”)
- The numeraire regions and products define a price indices. For the model, only the choice of the numeraire regions matter which enter the global factor price index, where the numeraire products and regions together define a producer price index currently not used in the equation setup.

Closures

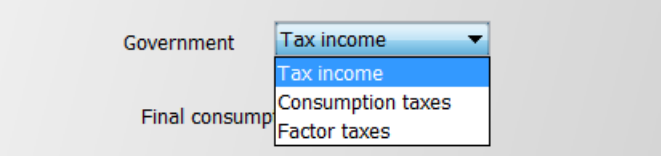


As shown above, the code supports alternative closures different from the ones used in the standard GTAP model:

- For **foreign savings**, besides the default global bank mechanism which leads equal expected returns to capital across all regions, the model can in run with a with allocation of investments and with fixed foreign savings. These closures are only available in the global model set-up.

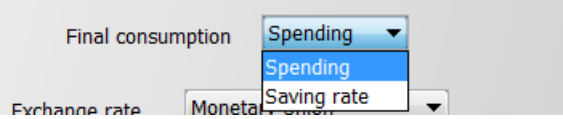


- The standard closure for the **government** is to calculate tax income at given tax rates, and to let the top level utility function allocate a certain share of regional income to government consumption. The reader should note that there is no immediate relation between changes in tax income and government consumption. Changes in government consumption relative to private consumption and regional saving depend on the one hand on changes of the related price indices and on the other on the elasticity of the private consumption share to changes in overall utility. As such, there is no closed government account, one could argue that implicitly government saving adjust. The alternative closures endogenize tax shifters either on all or selected products in consumption or for all or selected factors while keeping the real tax income for government consumption fixed, using the government price index to define real tax income. These two closures come closer to a representation of separate government account.

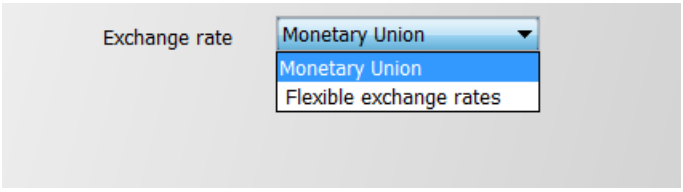


For the closures modifying taxes can be further detailed by applying the shifters

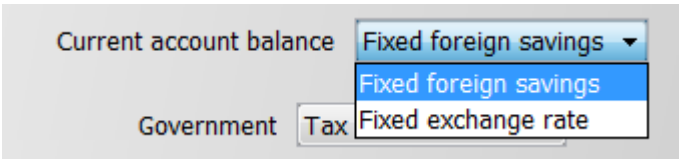
- The standard closure for **private consumptions and saving** is to a have flexible budget shares for private and government consumption and regional savings. The alternative closures allow to fix private consumption spending and to let the saving rate adjust.



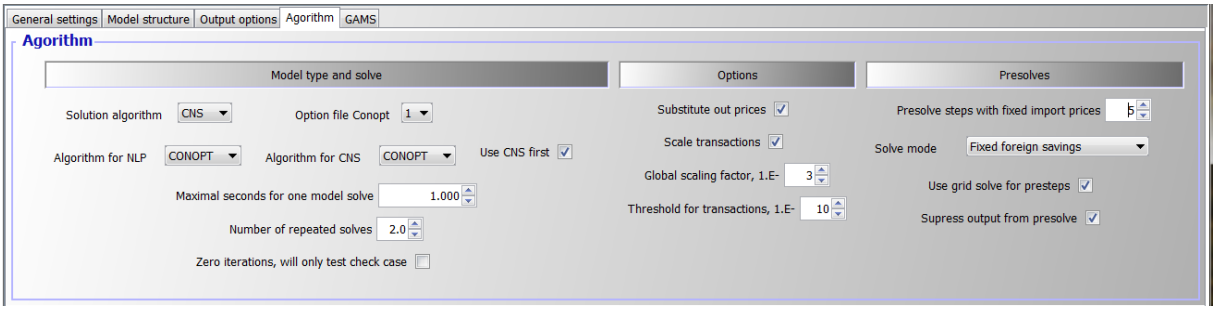
- The final closure relates to the **exchange rates**. In the standard GTAP model, there are no exchange rates and a world factor price index is used as the anchor (in the model, there is no money illusion such that only relative price matter). That is basically a set-up with a global monetary union. The alternative closure fixes the regional factor price indices and introduces flexible exchange rates for each region, as e.g. found in the GLOBE model. The reader should note that the two closures do **not** give identical results. These closures are only available in the global model.



- For the single region layout, two closures for the current account balances are offered: either fixed foreign saving with a flexible exchange rate or the reverse combination.



Algorithm



The model can be solved either as a constrained system of equations (formally solved with a dummy objective to yield a NLP which might help CONOPT) or as a MCP. The latter option is interesting if tax rates are to be endogenized e.g. under emissions ceilings, trade or production quotas, features not comprised in the standard model.

Please note that the default maximal time for model solution of 600 seconds might be too low for very large models.

The settings shown above are recommended defaults. Generally, solving the model as a constrained system of equations CNS - especially in connection with pres-solves, see below – has proven to be generally faster for larger models (> 150.000 transactions in SAM). The CNS solution will automatically check if the model is square. MCP gives additionally flexibility such as introducing Tariff Rates or production quota. Equally, the MCP version will automatically check for a unpaired equations and variables.

The standard option file is 1, which uses lower number of minor iteration, shown to speed up solution with larger models. The option file 2 with relaxed convergence tolerance is automatically chosen if the model cannot be solved in a first attempt. Option file 3 relaxes the convergence tolerance further, and is not intended for production runs, but for debugging purpose. Both with MCP and CNS, the above mentioned combinations of solver settings have proven to work best. The user can also chose which solver to use for NLP and CNS. Tests with a beta version of CONOPT4 which parallelizes e.g. matrix factorization has shown considerable speedups for very large models. The reader should however note the so far CONOPT4 is not officially released and that up to that point, only users highly familiar with the code of the model should use that beta version which shows great potential.

If the model is solved as a NLP, it will on demand first solved as a CNS. CONOPT uses a somewhat simpler and faster algorithm with CNS which might however sometimes fail. It is generally recommended to try “Use CNS first”.

Presolves

For large models, pre-solves can be used during which single region CGEs for each region in the model are solved independently. That process is repeated several times to inform the individual country CGE about changes in others regions. The single regions are only introduced as an intermediate steps towards the solution of the full model. They thus differ from regular single country CGEs which are usually solved either at fixed international prices or by rendering import and export prices depending on import and export quantities, not considering different trading partners. The pre-solves in here aim at providing a good starting point for a solve of the full global model.

Therefore, their structure needs to reflect the bi-lateral trade relations and other linkages between regions in the GTAP model. The layout of the single country models therefore differs somewhat from the usual single country CE structure and solely uses equations and variables already comprised in the model at unchanged parameterization.

In the standard GTAP model, there is no CET on the supply side. We therefore simply assume that a change in demand for bilateral exports of a single country has a negligible impact on the supply price of each importer. Accordingly, we drive the single country models with fixed fob prices on the import side. On the export side, however, we exploit the Armington structure: we feed changes in the country's supply prices in the Armington share equations of the importers at fixed total import and prices. The reader should note that solving the model also with fixed prices for exports would not help much in providing a good starting point for the model. Without a CET, the export price is by definition equal to the supply price in the domestic market. If one would hence fix the export prices, the domestic prices in the current solve would also be fixed. There would hence be no updated price information be passed along from solving one single country model to the next.

Further links in the full models between regions are based on the global factor price index which is fixed during a global model solve to reflect that there is no money solution, i.e. the behavioral equations are of degree zero in prices. However, during single model solves, fob prices are fixed such that it is impossible to fix at the same time the factor price index. In order to still drive the single models during iterations towards a global factor price index of unity, the fob prices are divided before each solve by the currently calculated global factor price index as are the saving price indx and the factor prices of all regions which enter fixed the single model solves.

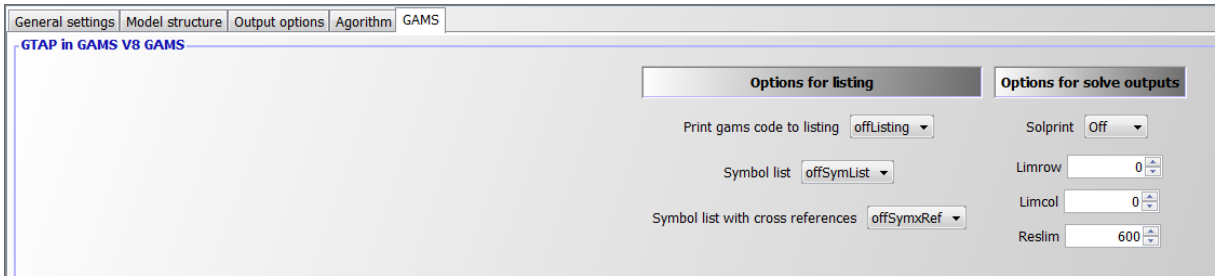
Finally, we reflect the global bank mechanism by calculating iterations the average expected return at global level, based on weighting with regional net investments. A heuristic estimate a change in foreign savings which should drive the country's expected rate towards the global average.

There are a number of options available for the pre-solves:

1. *The number of pre-solves*. For larger shocks and a higher number of regions, ten iterations are recommended as the pre-solves are relatively cheap.
2. *Use of grid solves*. That solves the single country models in parallel with the exemption of the last pre-solve, but exploits less information as results from other countries can only be exploited once all single countries are solved. The grid sole is the recommended option for modern multi-core machines.

CNS models based with CONOPT have proven to benefit much more from the pre-solves than the MCP version. Test have shown that the combination of presolves and CONOPT4 allow to solve 57x82 models without any filtering in about 15 minutes even under larger shocks.

GAMS



As all symbols (variables, equations) are also stored in a GDX container with GUI output switched on, one might normally switch “solprint” to “Silent” to keep the listing at minimal size.

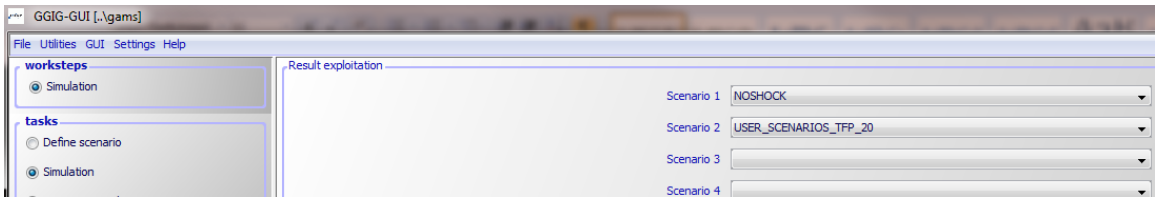
Inspecting the results

Before we turn to the exploitations tool of the GUI, the reader should be reminded that GTAPinGAMS allows for a wider range of possibilities to look at results:

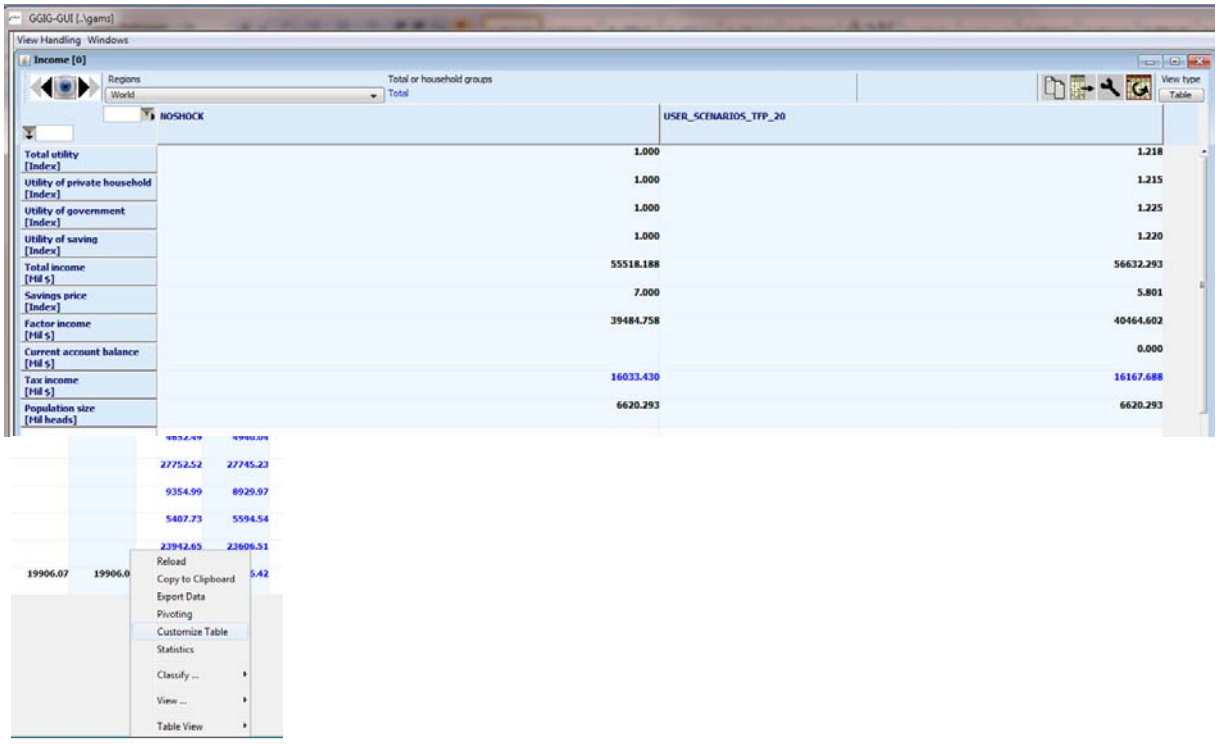
1. As with any GAMS model, solprint=on will produce a listing of the model solution showing variables and equations, see the GAMS documentation for details. That option can be switched on from the interface as discussed above.
2. If additionally, LIMROW and LIMCOL are non-zero, the individual equations will be shown with the Jacobian entries. The numbers will limit the output of individual instances shown and can be set by the GUI. For more information on how to interpret that output, please look at the GAMS manual.
3. If the output option GDX is switched on, all GAMS symbols (sets, parameters, variables) will be outputted to a GDX container and can be analyzed in a GDX viewer, either the one delivered with the GAMS IDE or the GDX viewer built into the GUI.
4. The model’s results can be mapped back into a SAM, and the SAM will be stored in the GDX.
5. The information from 1. -4. can also be inspected with the “Equation and Variable viewer” discussed in another section, which also provides a decomposition of each equation similar to “analzeGE”.
6. Finally, the exploitation tools can be used, which also cover the SAM, as discussed in the following.

Which option works best depends on the use case and user preferences. Model listings with “limcol” and “limrow” switched on are extremely useful during model development and debugging, whereas the exploitation tools are probably the best approach to systematically analyze results from a shock. Individual equation decomposition can complement both approaches. We can only recommend trying out all approaches at least once to find out which one fits best one’s preferences under specific use scenarios.

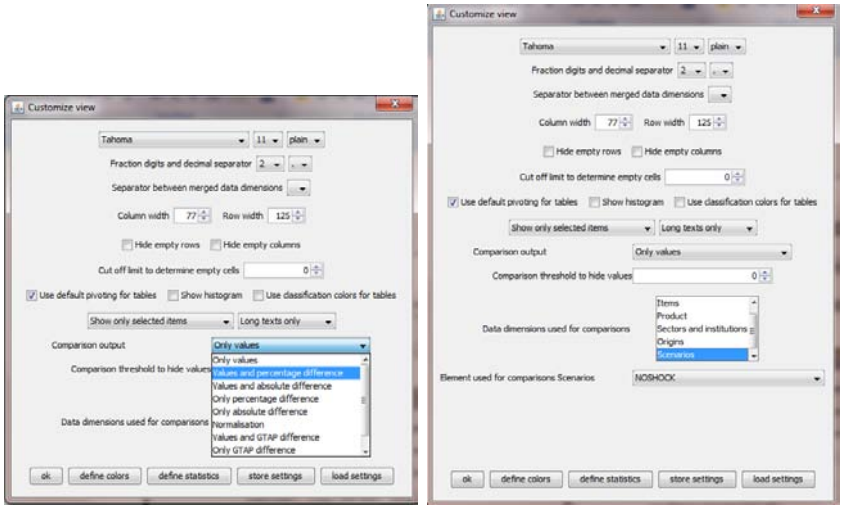
Inspecting results with GUI is nothing new: You have done that already if you followed the short introduction … so press on the “Exploit results” button and next on “Show results” again. Use the “NoShock” and your own scenario:



And now, you can see the counterfactual and the simulation side by side in the tables:



You might now want to add relative differences: click with the mouse in the table or use the "Customize" ( ) button in order to customize your view (see screenshot below). Choose "Values and percentage difference" as the "Comparison output", select "Scenarios" as the "Data dimensions used for comparisons" and make sure that the "NOSHOCK" scenario is used as the comparison element:

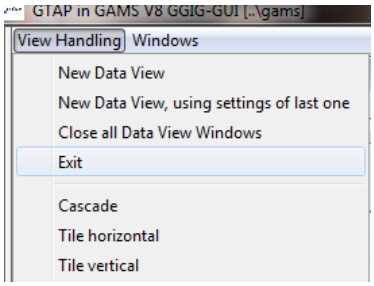


After closing the dialogue, you find now relative difference in small numbers beyond the simulated results in each cell of the table.

(Note: as expected, total utility increases, but prices drop …).

There are many other functionalities found in the exploitation tools, the GGIG user manual discusses these options: http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG_user_Guide.pdf.

If you want to leave the viewer again, use “Exit” from “View Handling” menu:

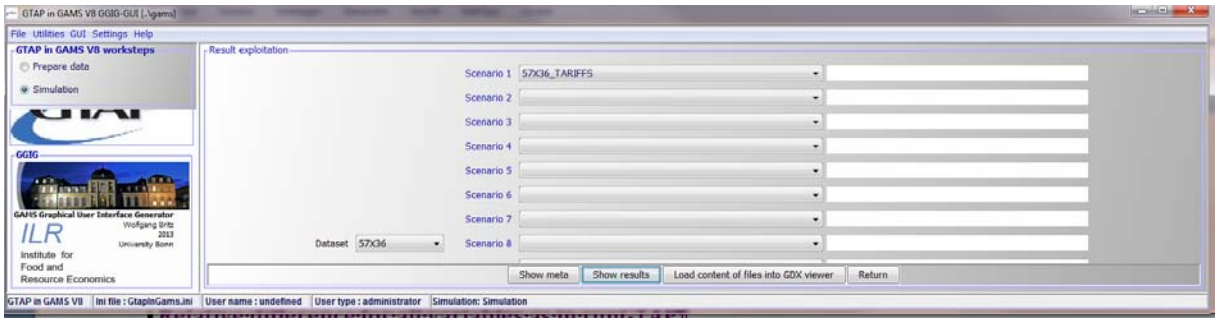


Notes for GEMPACK users or users familiar with other CGE models

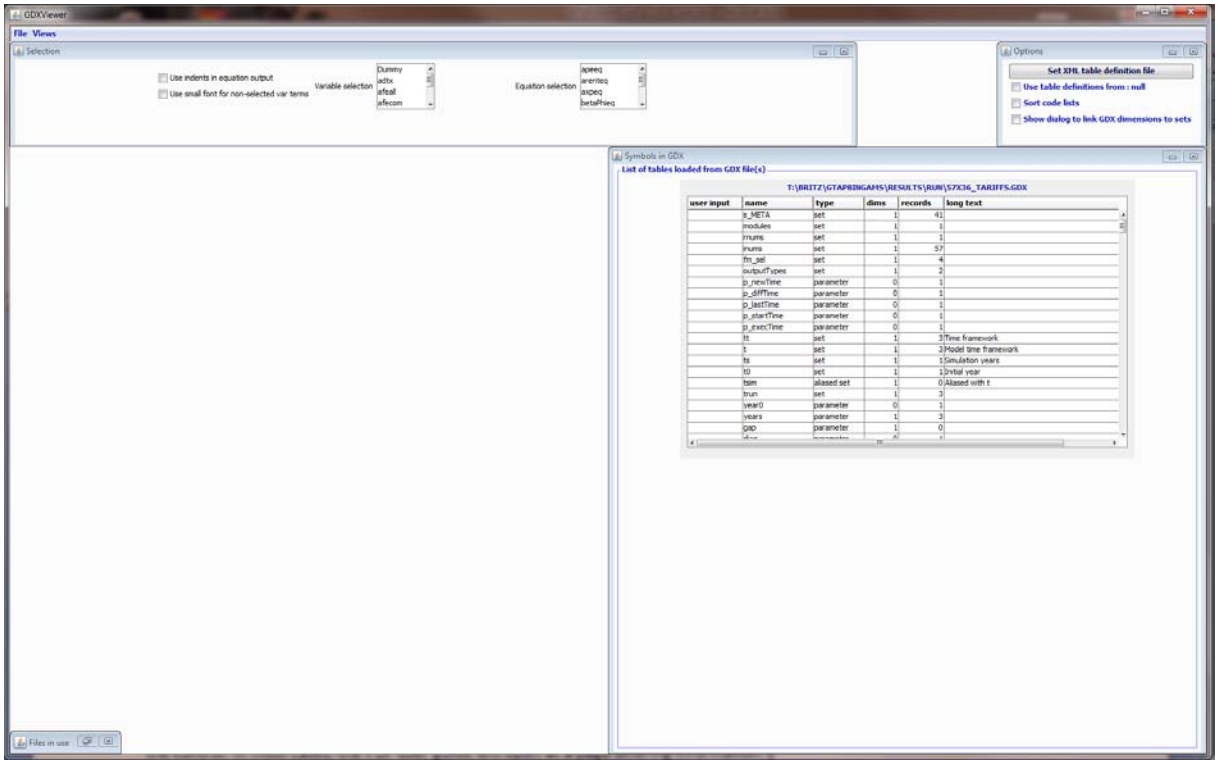
- The view “Model overview, SAM” shows a SAM generated from the model results (including row and column sums and potential imbalances) if the option “Store SAM in GDX” is switched on.
- Many tables comprise additional information on aggregates on top of the other rows. Only in some cases is what is called “total” a simple sum of the rows below. Thus, the information should generally not be confused with the “sum” information shown in “viewhar”.
- There is only in few selected cases a 1:1 relation between a variable in the model and numbers shown in table; typically the results shown in the different views stem from some post-model processing (e.g. multiplying a quantity index with a price).
- Instead of using the button, “Show results” which presents the different views discussed below, the “Load content of files in GDX viewer” allows to inspect all symbols used in the code, including all variables and the generated SAM (see next section). The GDX files are stored under “results\run” and can be naturally inspected with another GDX viewer such as the one comprised in the GAMSIDE.
- If the user want to inspect variables similar to “AnalyzeGE”, including decomposing equations, she can use the “Equation and variable viewer “ (see below).

Relative difference for all variables as in runGTAP

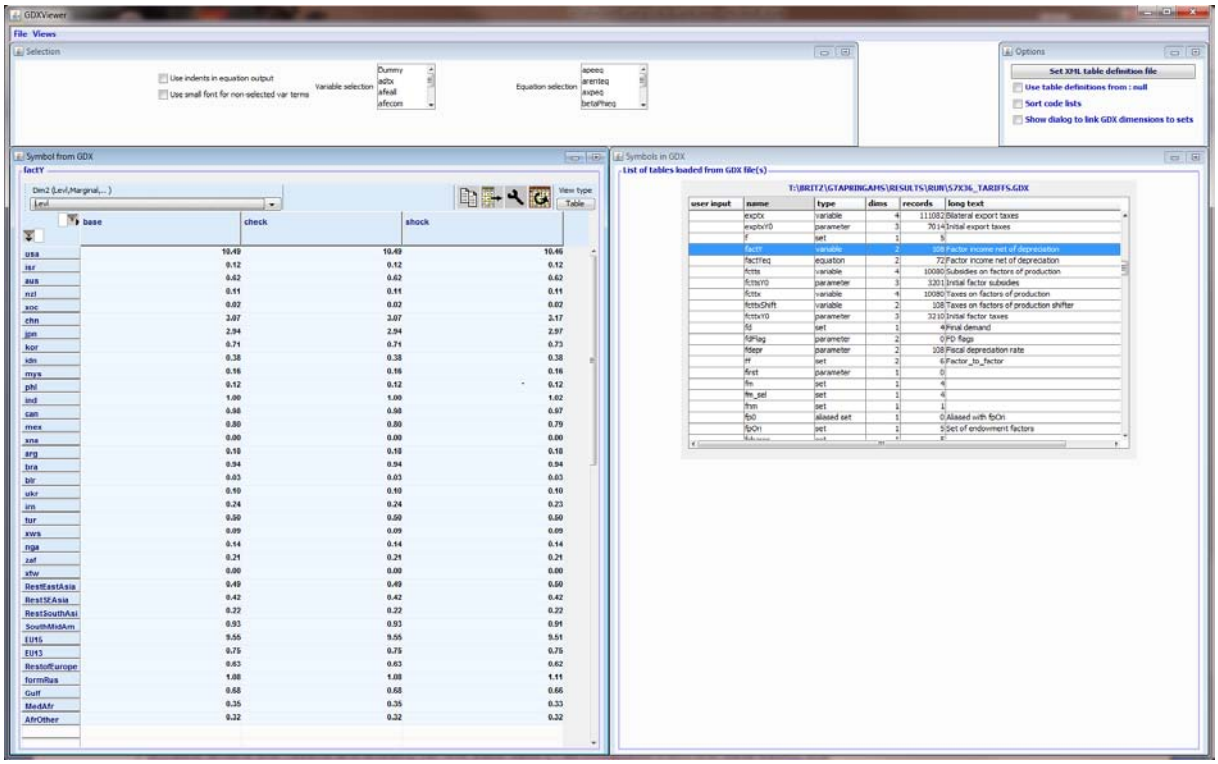
GEMPACK users typically compare simulated against benchmark levels for model variables. That is also possible with the interface. In order to do so, select the file with the result and press the “Load content of files into GDX viewer” button:



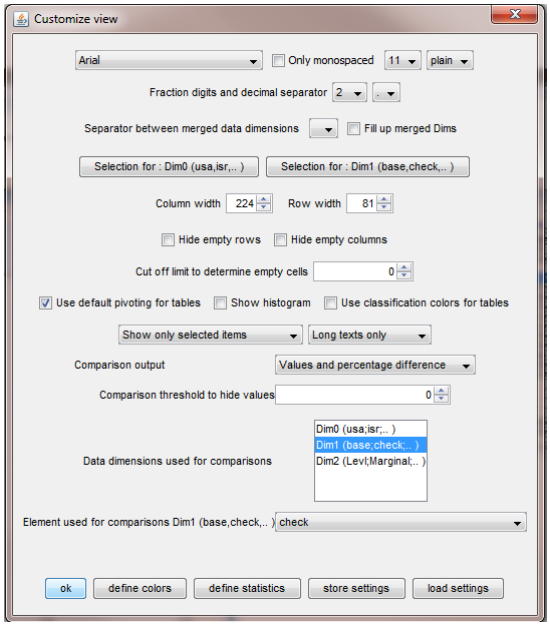
That will open a new window as shown below:



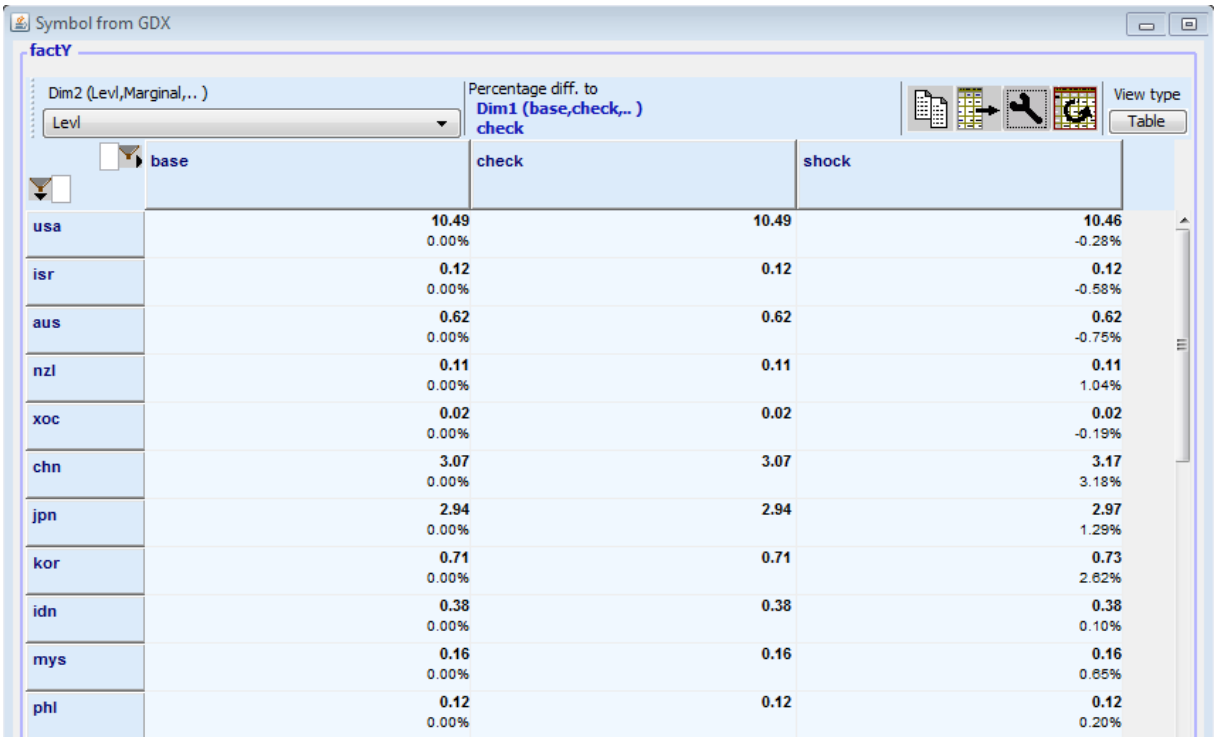
The window “List of table loaded from GDX file(s)” shows the status of all GAMS symbols (variable, equations, sets, parameters) after the solution of the shock. Double-click on any symbol you are interested in, e.g. the factor income “facty” as shown in the screen shot below. That will open a window where the results are shown, for the base case (= as initialized from the data base), the benchmark test (= check) and the shock.



In order to show percentage differences against the benchmark, open the option dialogue by pressing “ ”. Select e.g. “Values and percentage differences” under comparison output, chose the data dimension which end with “(check; …) and the item “check”:



The window with the result will now also comprises the information on the relative change as shown below:



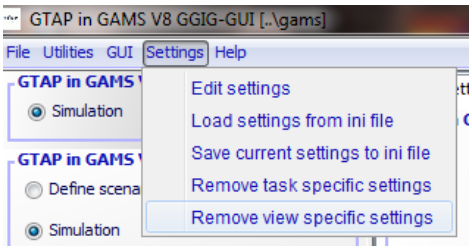
	base	check	shock
usa	10.49 0.00%	10.49	10.46 -0.28%
isr	0.12 0.00%	0.12	0.12 -0.58%
aus	0.62 0.00%	0.62	0.62 -0.75%
nzl	0.11 0.00%	0.11	0.11 1.04%
xoc	0.02 0.00%	0.02	0.02 -0.19%
chn	3.07 0.00%	3.07	3.17 3.18%
jpn	2.94 0.00%	2.94	2.97 1.29%
kor	0.71 0.00%	0.71	0.73 2.62%
idn	0.38 0.00%	0.38	0.38 0.10%
mys	0.16 0.00%	0.16	0.16 0.65%
phl	0.12 0.00%	0.12	0.12 0.20%

Trouble shooting with the viewer

If you are not sure what controls are for, try pressing the F1 button when the mouse is hovering over the control. In most cases, the PDF user guide will open at a page offering information.

Especially when working in the beginning with the viewer, one often ends up with a table showing no longer any data, an awkward selection, an unsuitable pivot etc.. The following strategies can be used to overcome such problems:

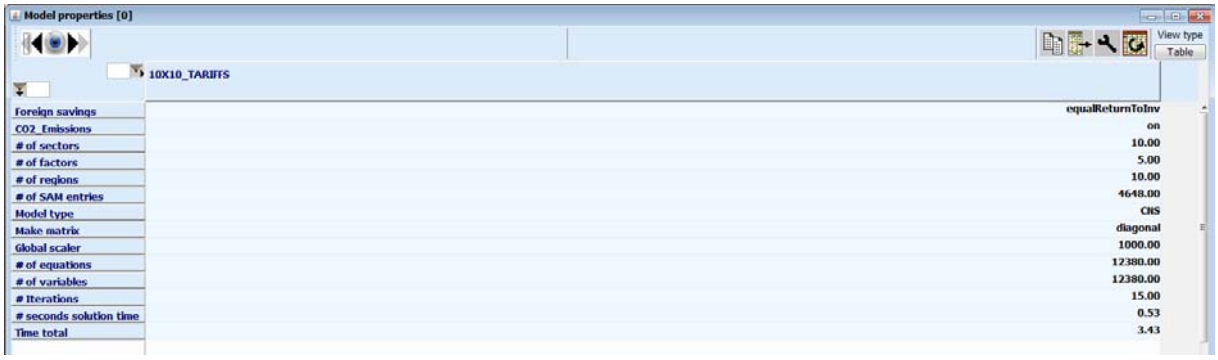
1. use the “close” button to remove the view, and use “View Handling”, “New Data View” to open a fresh one.
2. Leave the viewer (“View Handling”, “Exit”), and click on “Remove view specific settings” under “Settings” in the menu bar. That will set the viewer back to the original defaults.



3. If that does also not help, try to close and re-open the application.
4. If the viewer still shows curious things, it is most probably some programming error … contact the author.

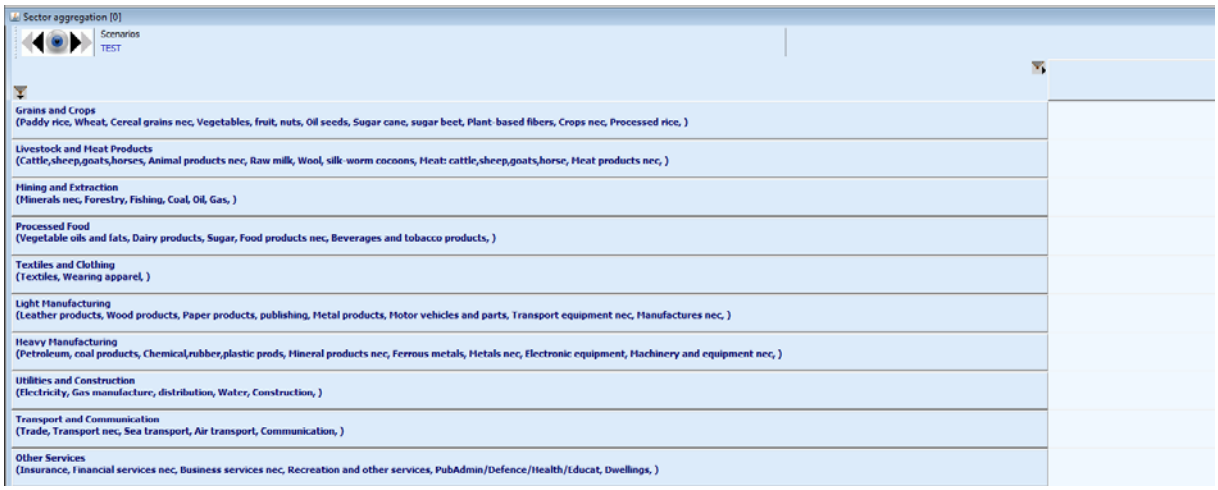
Overview on existing views for exploitation

Model overview/Model properties: reports some basic attributes of the model instance as seen below, such as the modules switched one, closures used or the size of the model:

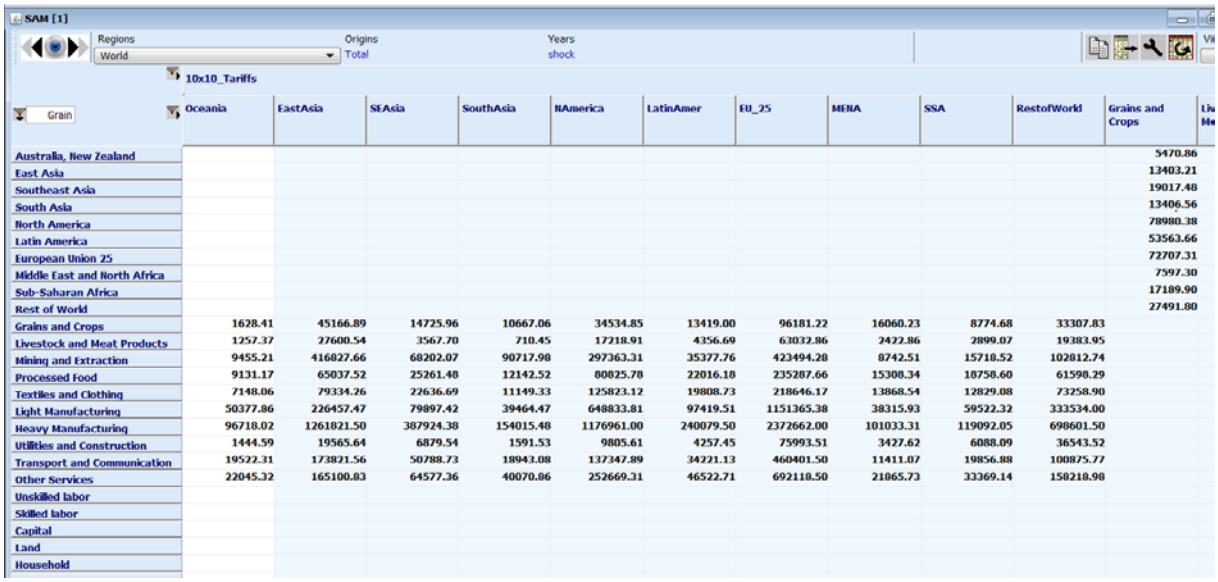


There are also two tables which report the aggregation of the data base under **Model overview/regional aggregation** and **Model overview/sectoral aggregation**:

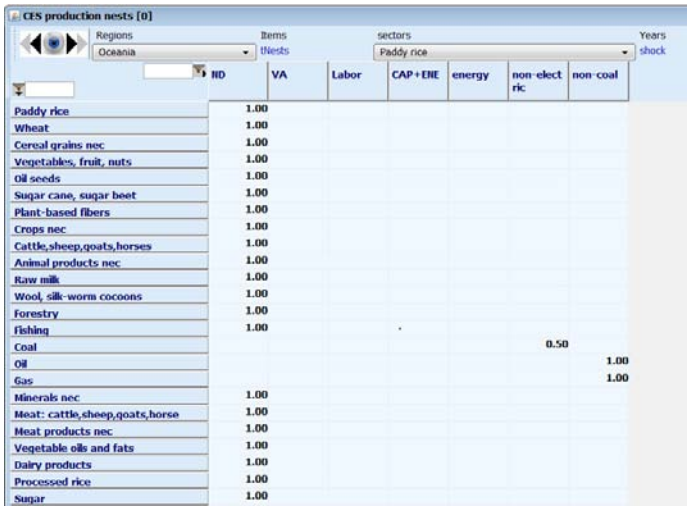
Regional aggregation [0]
Scenarios
TEST
Australia, New Zealand
(Australia, New Zealand, Rest of Oceania,)
East Asia
(China, Hong Kong, Japan, Korea, Mongolia, Taiwan, Rest of East Asia,)
Southeast Asia
(Cambodia, Indonesia, Lao People's Democratic Republ, Malaysia, Philippines, Singapore, Thailand, Viet Nam, Rest of Southeast Asia,)
South Asia
(Bangladesh, India, Nepal, Pakistan, Sri Lanka, Rest of South Asia,)
North America
(Canada, United States of America, Mexico, Rest of North America,)
Latin America
(Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Paraguay, Peru, Uruguay, Venezuela,
Rest of South America, Costa Rica, Guatemala, Honduras, Nicaragua, Panama, El Salvador, Rest of Central America, Caribbean,)
European Union 25
(Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece,
Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal,
Slovakia, Slovenia, Spain, Sweden, United Kingdom,)
Middle East and North Africa
(Rest of Western Asia, Egypt, Morocco, Tunisia, Rest of North Africa,)
Sub-Saharan Africa
(Benin, Burkina Faso, Cameroon, Cote d'Ivoire, Ghana, Guinea, Nigeria, Senegal, Togo, Rest of Western Africa,
Central Africa, South Central Africa, Ethiopia, Kenya, Madagascar, Malawi, Mauritius, Mozambique, Rwanda, Tanzania,
Uganda, Zambia, Zimbabwe, Rest of Eastern Africa, Botswana, Namibia, South Africa, Rest of South African Customs,)
Rest of World
(Oman, Israel, Switzerland, Norway, Rest of EFTA, Albania, Bulgaria, Belarus, Croatia, Romania,
Russian Federation, Ukraine, Rest of Eastern Europe, Rest of Europe, Kazakhstan, Kyrgyztan, Rest of Former Soviet Union, Armenia, Azerbaijan, Georgia,
Bahrain, Iran Islamic Republic of, Kuwait, Qatar, Saudi Arabia, Turkey, United Arab Emirates, Rest of the World,)


Sector aggregation [0]
Scenarios
TEST
Grains and Crops
(Paddy rice, Wheat, Cereal grains nec, Vegetables, fruit, nuts, Oil seeds, Sugar cane, sugar beet, Plant-based fibers, Crops nec, Processed rice,)
Livestock and Meat Products
(Cattle,sheep,goats,horses, Animal products nec, Raw milk, Wool, silk-worm cocoons, Meat: cattle,sheep,goats,horse, Meat products nec,)
Mining and Extraction
(Minerals nec, Forestry, Fishing, Coal, Oil, Gas,)
Processed Food
(Vegetable oils and fats, Dairy products, Sugar, Food products nec, Beverages and tobacco products,)
Textiles and Clothing
(Textiles, Wearing apparel,)
Light Manufacturing
(Leather products, Wood products, Paper products, publishing, Metal products, Motor vehicles and parts, Transport equipment nec, Manufactures nec,)
Heavy Manufacturing
(Petroleum, coal products, Chemical,rubber,plastic prods, Mineral products nec, Ferrous metals, Metals nec, Electronic equipment, Machinery and equipment nec,)
Utilities and Construction
(Electricity, Gas manufacture, distribution, Water, Construction,)
Transport and Communication
(Trade, Transport nec, Sea transport, Air transport, Communication,)
Other Services
(Insurance, Financial services nec, Business services nec, Recreation and other services, PubAdmin/Defence/Health/Educat, Dwellings,)

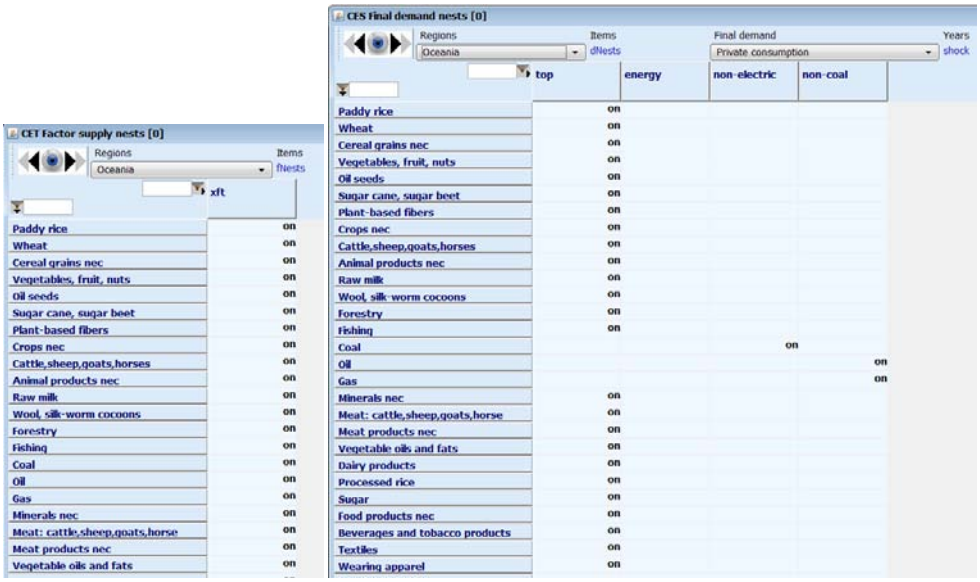
Model Overview/SAM reports the results in a SAM, including row and columns sum and potential imbalances. Due to rounding errors and feasibility tolerances of the solvers, small imbalances in relative terms are possible and not of concern:



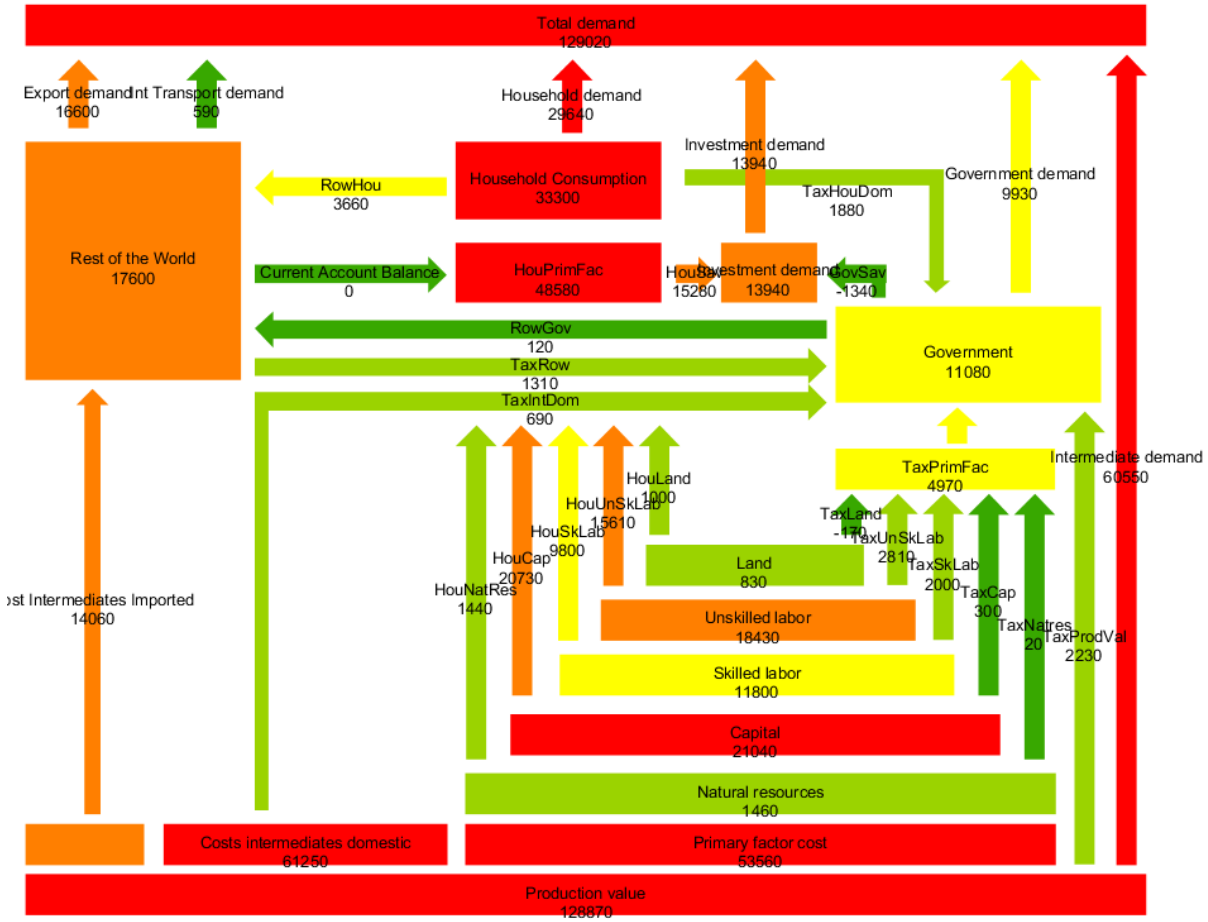
Model Overview/CES products nests reports the nesting structure used in the production module:



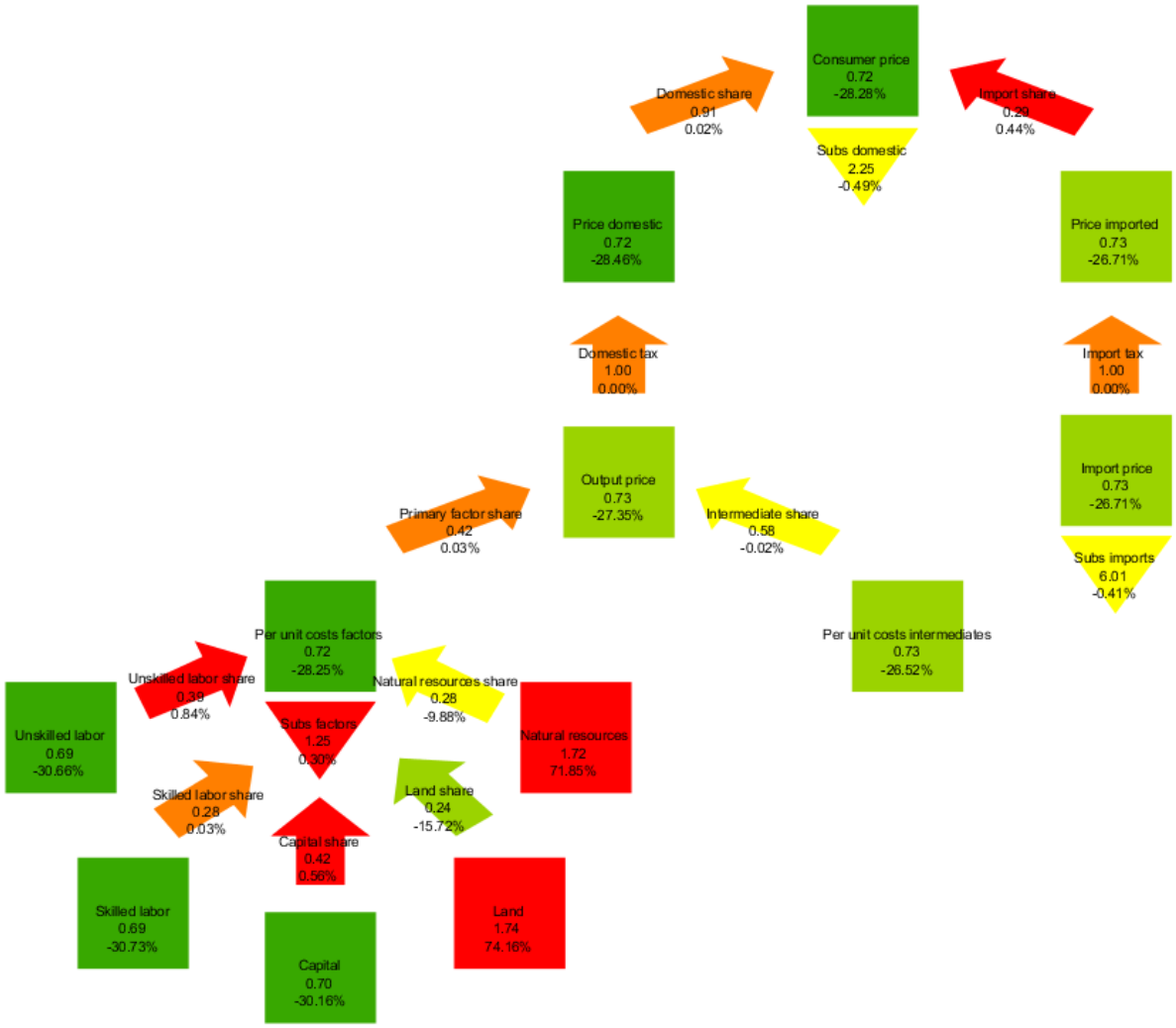
Similar views are available for the nesting used for factor supply and final demand:



Model Overview/Flow schemer uses the in-built mapping utility in a somewhat unusual way to draw a flow chart with all transactions in the economy, some aggregated. The lowest rectangle shows the total production value; the related cost are split into intermediate (imported and domestic) demand on the LHS rectangles above, and total primary factor costs on the RHS rectangle above. Total primary factor income remuneration for each factor is split up into tax (arrows on right) and household income (arrows to the left). The rectangle “Government” shows the government account; incoming arrows show tax income, outgoing ones government spending (for domestic, imports and saving; the latter is residually calculated). The rectangle “Household consumption” shows final consumption by the household, the arrow to its left shows spending for imports net of taxes. The rectangle on the top is final demand; it is sourced by export and international transport demand (left), household, saving and government demand, and domestic intermediate demand for that sector. The schemer view, as with any view, can also show differences between scenarios and provides an aggregation to total world. The schemer can also be used to visualize individual sectors and commodities, or an aggregation to total output/demand.


Total demand
129020
Export demand
16600
Int Transport demand
590
Household demand
29640
Investment demand
13940
Government demand
9930
Rest of the World
17600
RowHou
3660
Household Consumption
33300
TaxHouDom
1880
Current Account Balance
0
HouPrimFac
48580
HouSave
15280
Investment demand
13940
GovSav
-1340
RowGov
120
TaxRow
1310
TaxIntDom
690
Government
11080
TaxPrimFac
4970
Intermediate demand
60550
HouLand
1000
HouUnSkLab
15610
HouSkLab
9800
HouCap
20730
HouNatRes
1440
TaxLand
-170
TaxUnSkLab
2810
TaxSkLab
2000
TaxCap
300
TaxNatres
20
TaxProdVal
2230
Land
830
Unskilled labor
18430
Skilled labor
11800
Capital
21040
Natural resources
1460
Costs intermediates domestic
61250
Primary factor cost
53560
Production value
128870

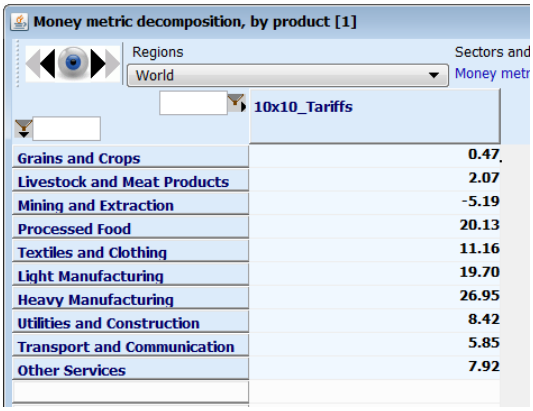
Similarly, **Model Overview/Price schemer** visualizes the composition of per unit costs and prices, where the boxes represent price, the arrows show costs shares respectively tax base multipliers and the triangles substitution elasticities. Combined, as shown below, with a percentage difference to the baseline, it allows a rather straightforward interpretation of major price changes in the model. The results are available per region and sector/commodity, as well as aggregated to total output respectively private consumption and world level.

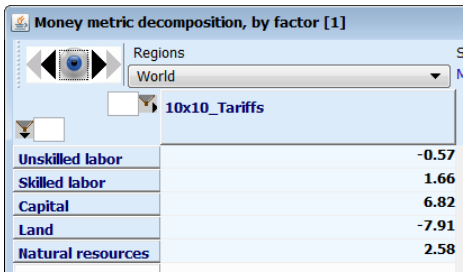


The table **Welfare/Income** provides an overview in key results such as GDP and price indices**:**

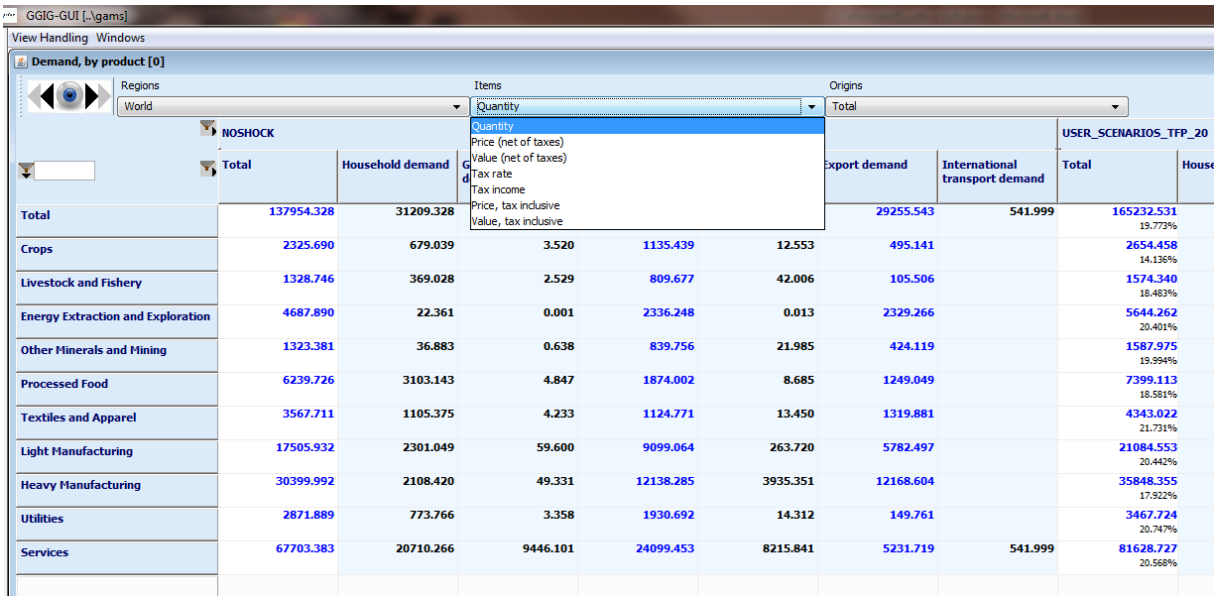
The three tables **Welfare/Money metric decomposition, Welfare/Money metric decomposition, by product** and **Welfare/ Money metric decomposition, by factor** support an welfare analysis. "By products" reports welfare gains or losses related to output price changes, by factor linked to changes in factor prices or quantities:

Money metric decomposition [1]
Regions
World
10x10_Tariffs
Total [Bio $] 33.32
Output price index [Bio $] 155.10
Private price index [Bio $] 111.53
Government price index [Bio $] 7.66
Savings price index [Bio $] 35.65
Income total [Bio $] -121.63
Factor income [Bio $] 2.58
Indirect tax income [Bio $] -147.84

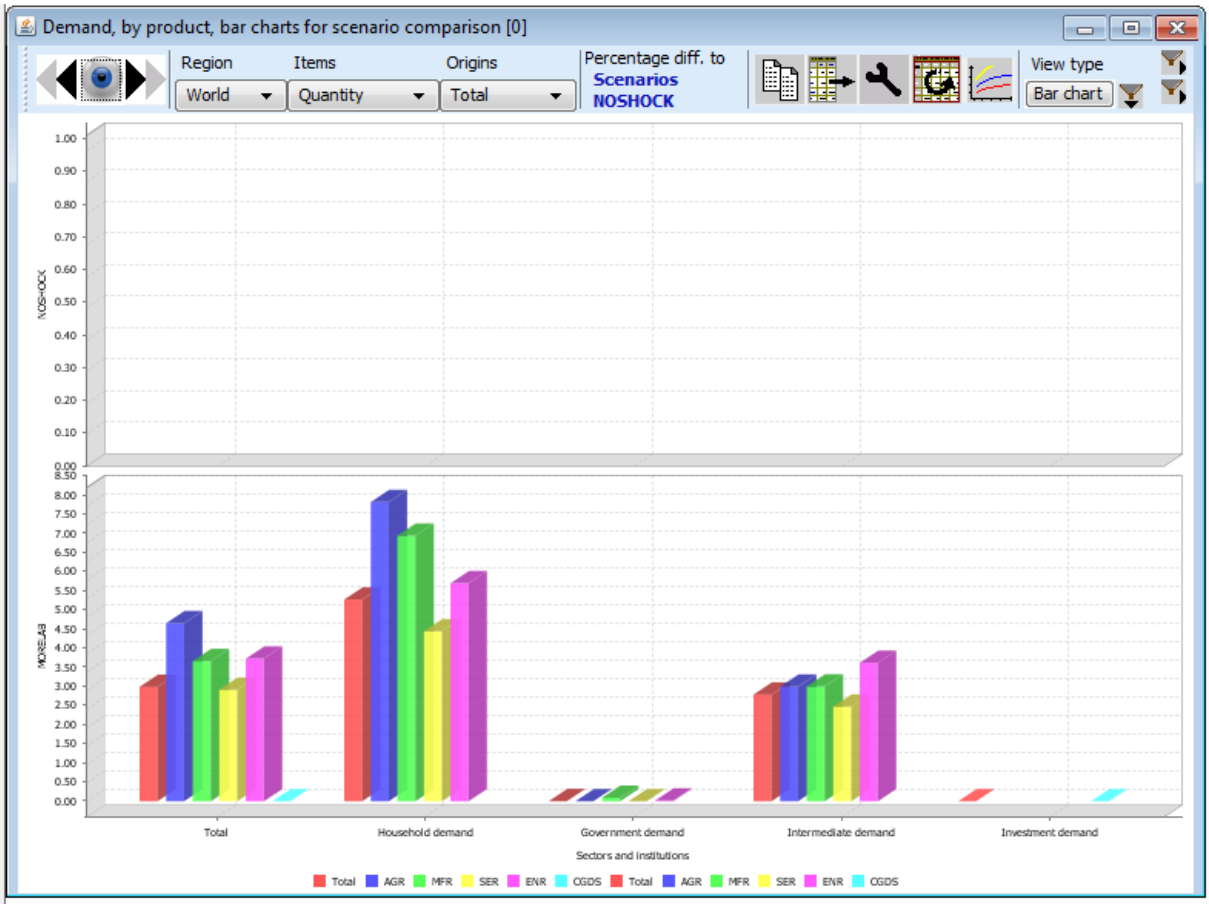




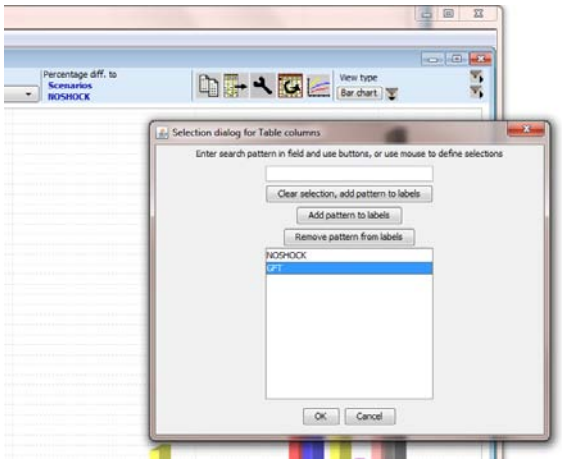
Markets/Demand by product: Shows a table with demand by institutions, products and totals shown in the row. The items box allows for looking at quantities, prices, tax rates, value and tax income, the origins by "domestic", "imported" and "total". The first box allows selecting the regions. Note: total is an aggregation over all products.



A predefined view "**Demand, by product, bar charts for scenario comparison**" as seen below shows how the combination of a certain pivot, view type while using the comparisons can be used to produce bar charts showing relative differences. It might be useful to the filter out the comparison scenario from the view such that the empty upper graph is not shown.

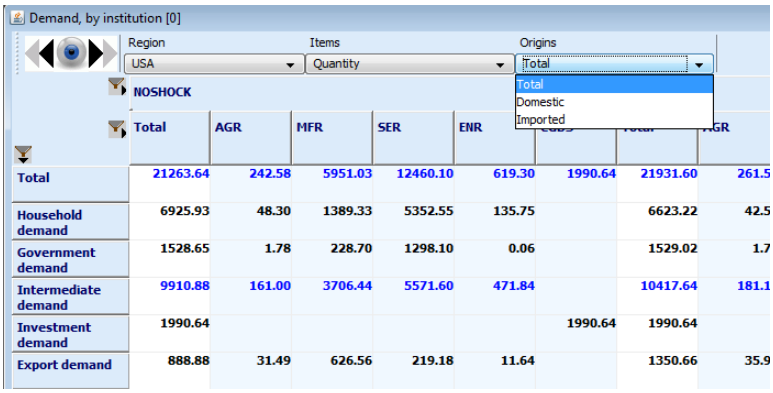


Note: you use the second "filter" bottom next to view type to "view type" (double click on it) to open a dialogue which lets you remove the benchmark scenario:



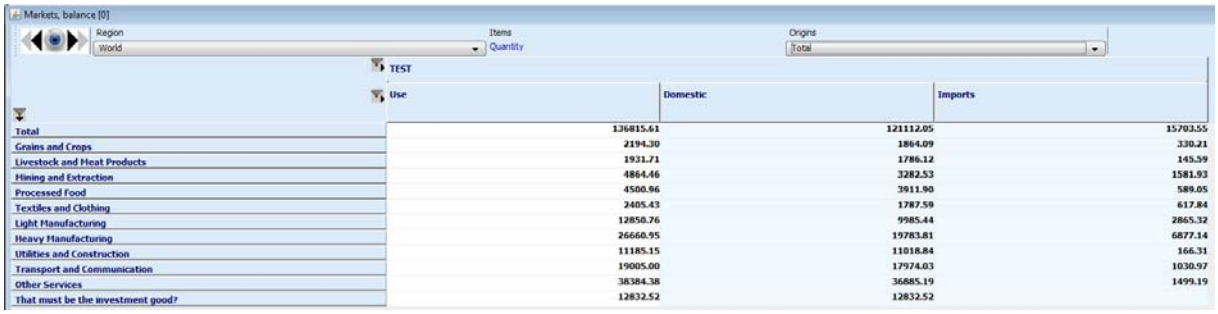
That will produce a nicer graph.

Markets/Demand by institution: same content as before, but different pivot (could also be quickly produced by using the pivot possibility which can be applied to any view)



Markets/Final demand nests: shows the demand aggregates –quantities, prices and values – according to the active demand nesting:

Markets/Balance: Shows total use and how it is sourced:



Demand, per product, per capita in $ is an example for a tabled definition which uses the in-built expression evaluator, it divides total demand by the population size:

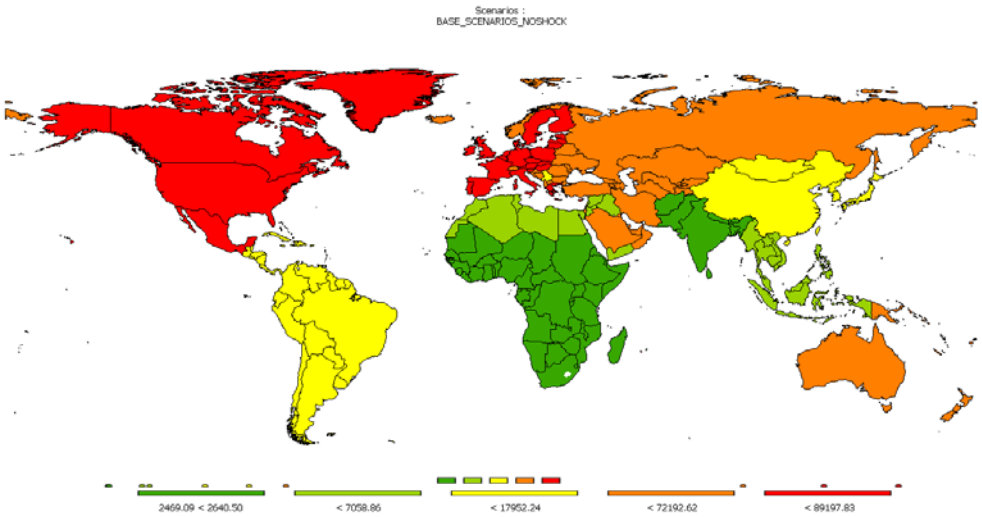
Demand, by product, per capita in $ [0]

Region: World | Items: Q | Origins: Total | View type: Table

BASE_SCENARIOS_NOSHOCK

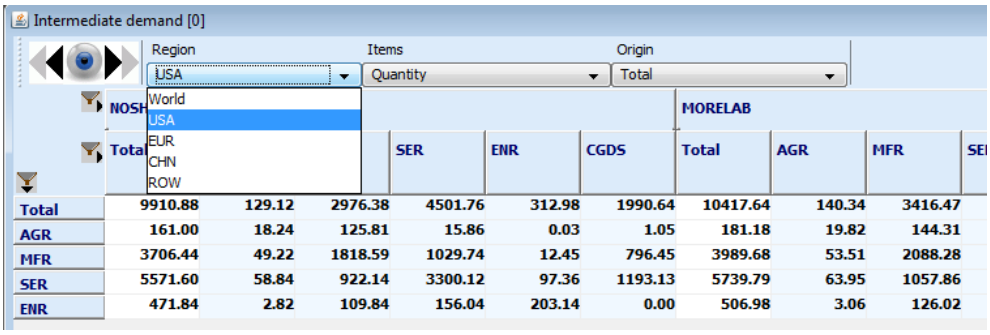
	Total	Household demand	Government demand	Intermediate demand	Investment demand	Export demand
Total	19660.35	4602.31	1424.48	9700.63	1938.36	1924.02
Grains and Crops	217.49	89.83		111.79		15.87
Livestock and Meat Products	212.19	101.69		100.34		10.15
Mining and Extraction	638.36	7.88		444.66		185.82
Processed Food	635.65	372.83		191.11		71.71
Textiles and Clothing	244.44	103.20		89.15		52.09
Light Manufacturing	1901.86	352.23	4.66	1157.64		387.32
Heavy Manufacturing	3883.43	345.09	2.69	2614.66		920.99
Utilities and Construction	1635.37	125.83	1.69	1507.48		0.37
Transport and Communication	2765.58	1183.57	15.02	1371.13		125.32
Other Services	5587.62	1920.16	1400.42	2112.67		154.38
That must be the investment good?	1938.36				1938.36	

That view can alternatively be visualized as a map in **Demand, per product, per capita in $, map:**

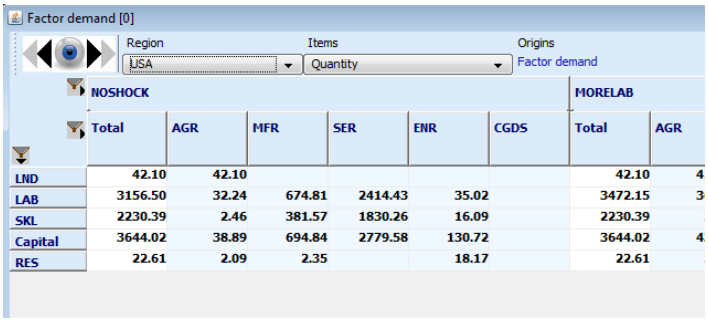


Maps can be made from any table by putting the regions in the rows and choosing “Map” under “View type”.

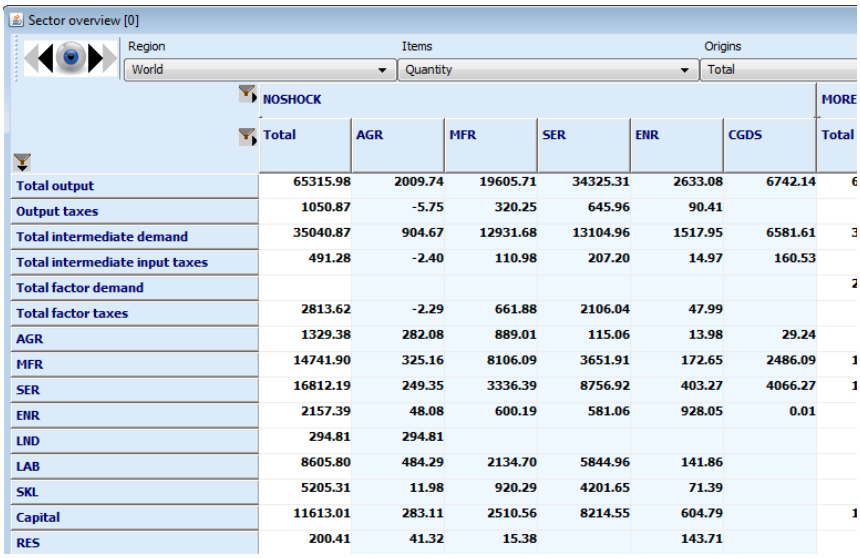
Markets/Intermediate demand: break of the intermediate demand by sector. Note: the total in the columns is an aggregation over all sectors.



Markets/Factor demand: report primary factor use, price etc. by sector



Sectors/Sector overview: reports for each sector total output, output taxes, total intermediate and total factor demand, as well the individual intermediate and factor demands.



Sectors/Sector cost shares: a selection of the value item of the view above, normalized by total output:

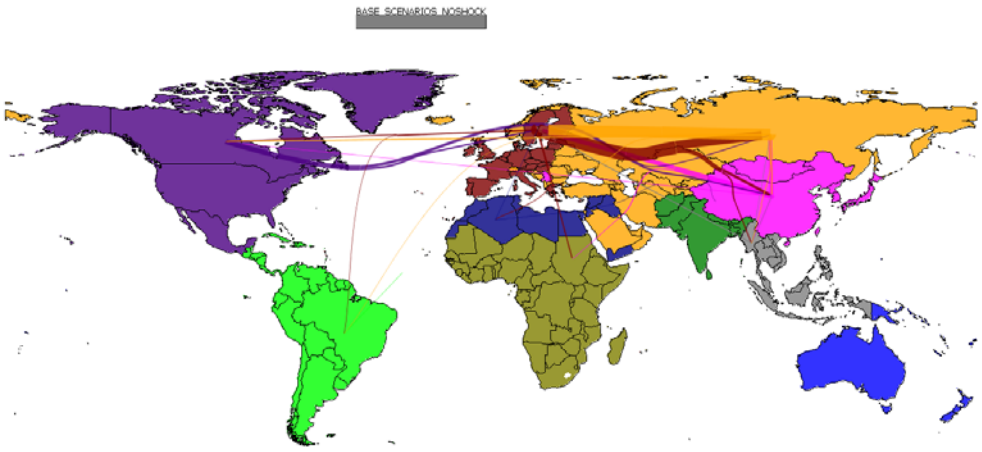
Cost shares [0]

Region: World | Items: Value | Origins: Domestic

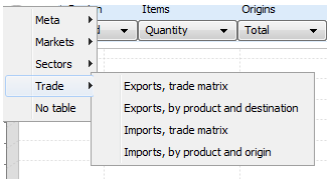
NOSHOCK

	Total	AGR	MFR	SER	ENR	CGDS
Total output	1.00	1.00	1.00	1.00	1.00	1.00
Total intermediate demand	0.45	0.40	0.52	0.34	0.42	0.82
Total intermediate input taxes	0.01	-0.00	0.00	0.01	0.00	0.02
Total factor demand						
Total factor taxes	0.04	-0.00	0.03	0.06	0.02	
AGR	0.02	0.13	0.04	0.00	0.01	0.00
MFR	0.16	0.13	0.30	0.09	0.04	0.22
SER	0.24	0.12	0.16	0.24	0.14	0.60
ENR	0.03	0.02	0.03	0.02	0.22	0.00
LND	0.00	0.15				
LAB	0.13	0.24	0.11	0.17	0.05	
SKL	0.08	0.01	0.05	0.12	0.03	
Capital	0.18	0.14	0.13	0.24	0.23	
RES	0.00	0.02	0.00		0.05	

There are further tables available to bi-lateral trade which can also be visualized as flow maps as shown in the screen shot below,



and more might be added in the future. But the examples above might be sufficient to judge if these views (in addition of using a GDX viewer to look out variables) are a useful addition to result analysis.



Sensitivity analysis

GEMPACK supports systematic sensitivity analysis (Channing 1996). The standard GTAP in GAMS model also offers similar functionality; drawing on experiences with other GAMS-based models. In order to keep the technical solution relatively easy, the following approach is chosen:

1. The GAMS program which is used to run a single shock is also used for single sensitivity experiments, in order to use as much as possible existing code and to avoid duplicating code fragments.
2. A second GAMS program in combination with utility from SANDIA labs[6] defines the experiments and executes in parallel GAMS children, which solve the same scenario using each its own parameter settings.
3. That second GAMS mother program collects the results from the children and combines them into one GDX file.

Currently, the sensitivity analysis allows changing substitution and transformation elasticities by defining *relative* ranges around the given parameters, plus generates on demand additional draws for two "shocks", which needs to be translated by changes in the GAMS code into something useful. These relative ranges need not be symmetric. The draws can be defined generally in two modes:

1. The same relative change is applied simultaneously for all sector/factors/shocks. To give an example: if the drawn relative change for the factor transformation is 0.5, all transformation elasticities are reduced by 50%.
2. Individual draws for each sector and factor are generated.

Whereas Arndt 1996 uses Gaussian Quadratures, the approach in here is based on a combination of Latin Hypercube Sampling (LHS) and an entropy estimator. The user chooses the number of draws, and the LHS uses stratified random sampling for space filling design. The LHS is chosen as it can be easily applied also in case of non-symmetric, truncated distributions. In the default case, the LHS simply returns (more or less) uniform distributed draws across the factor range, i.e. between the relative lower and upper limit chosen by the user, with an average presenting the mean between the lower and upper limit.

If the user has chosen non-symmetric bounds around zero, the simulations would hence lead to a systematic deviation from the mean parameter used without sensitivity analysis. The entropy estimator can therefore be used to determine probability weights for the draws which ensure that the weighted average over the draws for each relative factor change is close to unity. That approach is hence quite similar to Gaussian Quadrature which also applies weights to the observations. The difference here is that a uniform distribution over the chosen range is the preferred distribution, while the Gaussian Qudratures used in GEMPACK tries to mimic normal or triangular distributions.

The user has the choice between *three difference distributions*, all truncated at the chosen lower and upper limit:

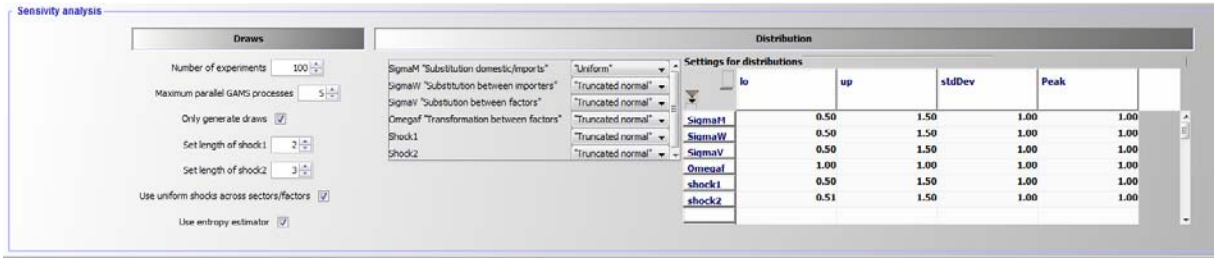
1. Uniform
2. Truncated Normal distribution
3. Triangular distribution

For the truncated Normal, the standard deviation must be set, for the triangular distribution the peak. The combination of user set truncation points and distributions allows for a rather flexible design of the experiments. The program can also easily be extended to also apply sensitivity analysis to shocks.

[6] http://www.sandia.gov/

The general set-up of the sensitivity analysis is defined on the left-most group of controls: the number of experiments (= parameter draws and related model solves), the maximum number of parallel solves.

Controls for sensitivity experiments



The left hand side determines the number of draws and how many GAMS processes are used in parallel to simulate the different experiments. The user might also chose “Only generate draws” to prevent the actual model runs, in order to check the layout of the scenarios.

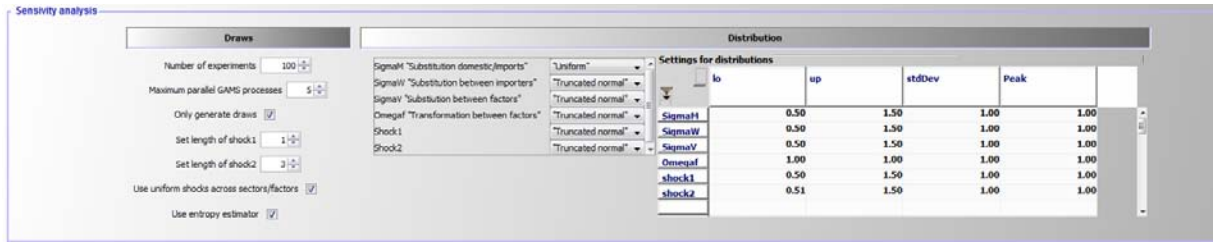
Using uniform shocks across sector/factors will apply the same relative change for the chosen parameters to all sectors and factors. Otherwise, each sector / factor will receive its own random shock in each experiment. The panel on the right hand side allows choosing the type of distribution for each parameter and the related settings. The “Peak” defines the mode for the triangular case.

Entropy estimator

In the case of non-symmetric truncation points and use of triangular or normal distributions, as in the last example above, the mean of the draws will deviate from unity. On demand, an entropy estimator will try to find probability weights for the draws which recover a mean of unity. If no solution is found, the estimator tries bounds around each mean of +/-1%, +/- 2.5% and +/-5%. If that fails, the program will abort with an error.

Without the entropy estimator, each draw receives an equal weight and the mean relative parameter draws might deviate from unity.

Drawing systematic experiments for other elements of the simulation



As seen from above, the interface allows defining two “shocks” with related set lengths. The simplest layout is shown above: one shock with a set length of unity. The automatically generated include files will each carry lines as follows:

```
parameter shock1(*);set s1/s1*s1/;
 shock1('s1') =1.4839095406;
```

The shock1 parameter can be used to e.g. change the population size in all regions by the same factor. If regional specific shocks are needed, the length of the set must be equal or higher than the

number of regions. Assume e.g. a dis-aggregation with 5 regions, and the set length of shock1 accordingly increased to 5. The include file will comprise lines as follows:

```
parameter shock1(*);set s1/s1*s5/;
 shock1('s1') =0.6891764639;
 shock1('s2') =0.7597044235;
 shock1('s3') =0.6128377057;
 shock1('s4') =1.2482232299;
 shock1('s5') =1.3406050177;
```

With a cross set between r and s1, using the **pos** operator from GAMS, the numerical values on shock1 can be mapped e.g. on the population size:

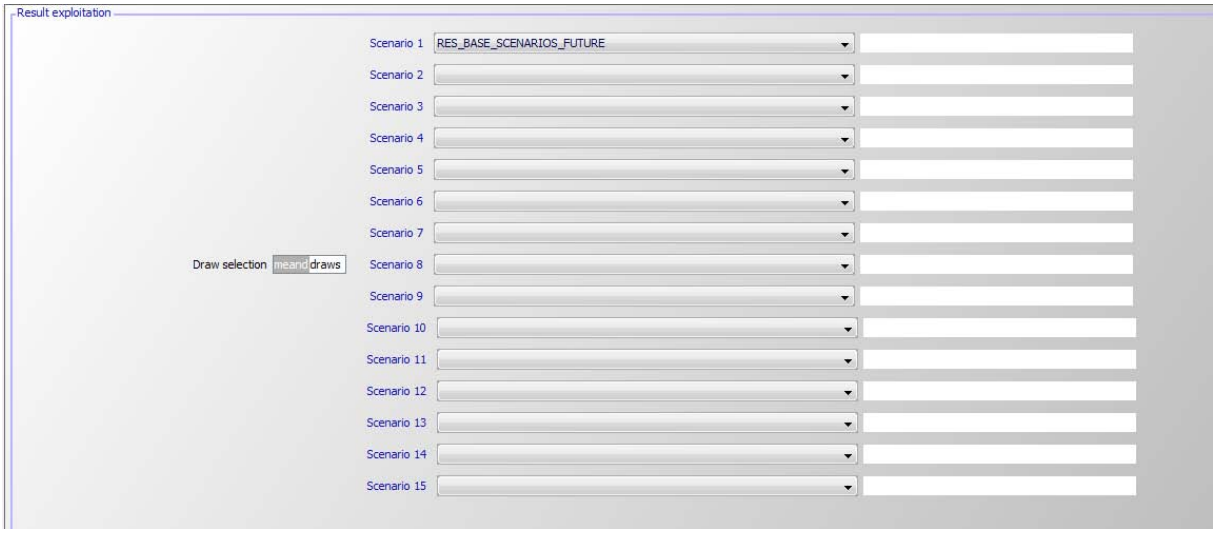
```
set r_to_s1(r,s1);

r_to_s1(r,s1) $ (r.pos eq s1.pos) = yes;
pop(r) = pop(r) * sum(r_to_s1(r,s1), shock1(s1));
```

The two shocks in combination with flexible set lengths should hence allow drawing from experiments for different types of shocks quite easily without the need to change the GAMS code for the sensitivity analysis. Note that these shocks can be combined with changes to the parameters.

Parallel solves and analyzing the results

The actual solves for each individual experiments start from a savepoint generated after solving the benchmark, such that execution time is only spend on model generation, solving and post-model reporting. The solve times reported above stem from a by now medium fast four-core desktop and underline that from a computational view point, sensitivity analysis is no longer a concern.

The interface will allow showing either all draws and/or the weighted average of the draws (meand):



If only the mean over the draws is selected, the result tables look almost identical to a single simulation run. The tables will report two outcomes: (1) the mean of the draws labeled “meand” and the (2) result at the original parameter “mean” – what one would simulate without any sensitivity analysis:

Demand, by product [0]

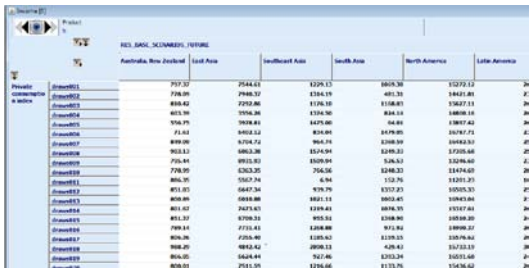
Region: World | Items: Value | Origins: Total | Percentage diff. to: Draws mean

RES_USER_SCENARIOS_GFT

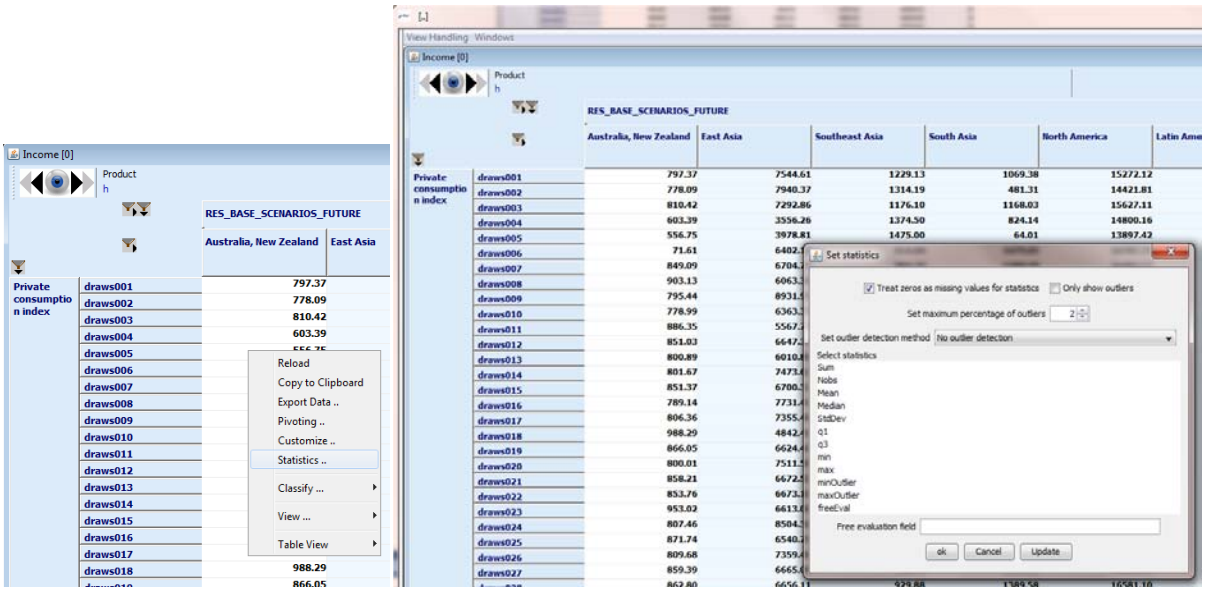
		Total	Household demand	Government demand	Intermediate demand	Investment demand	Export demand	International transport demand
Total	meand	131296.89 -0.02%	30592.51 -0.02%	9509.66 -0.02%	64336.11 -0.03%	12834.53 -0.02%	13503.61 -0.03%	520.48 -0.04%
	mean	131327.41	30598.80	9511.28	64352.24	12837.40	13507.00	520.67
Grains and Crops	meand	1462.23 -0.13%	591.16 -0.12%		751.72 -0.13%		119.35 -0.10%	
	mean	1464.07	591.89		752.71		119.47	
Livestock and Meat Products	meand	1410.77 -0.05%	674.03 -0.08%		665.66 -0.04%		71.07 0.12%	
	mean	1411.47	674.56		665.91		70.99	
Mining and Extraction	meand	4261.06 -0.09%	51.20 -0.14%		2871.75 -0.10%		1338.11 -0.07%	
	mean	4264.97	51.27		2874.67		1339.03	
Processed Food	meand	4244.36 -0.02%	2477.29 -0.03%		1248.83 -0.02%		518.24 -0.01%	
	mean	4245.42	2478.05		1249.07		518.30	
Textiles and Clothing	meand	1744.34 -0.08%	683.58 -0.04%		642.55 -0.15%		418.20 -0.03%	
	mean	1745.66	683.84		643.49		418.33	
Light Manufacturing	meand	12733.19 -0.01%	2332.11 -0.01%	29.84 -0.04%	7643.46 -0.01%		2727.78 -0.01%	
	mean	12734.61	2332.41	29.86	7644.27		2728.07	
Heavy Manufacturing	meand	25936.77 -0.03%	2292.69 -0.02%	17.20 -0.04%	17176.41 -0.02%		6450.47 -0.03%	
	mean	25943.26	2293.26	17.21	17180.67		6452.12	
Utilities and Construction	meand	10878.77 -0.03%	834.30 -0.04%	11.21 -0.12%	10030.96 -0.02%		2.31 -0.38%	
	mean	10881.54	834.60	11.22	10033.40		2.32	
Transport and Communication	meand	18524.80 -0.02%	7878.17 -0.02%	100.56 -0.04%	9187.13 -0.02%		838.46 -0.01%	520.48 -0.04%
	mean	18528.80	7879.68	100.60	9189.25		838.58	520.67
Other Services	meand	37266.07 -0.01%	12777.97 -0.01%	9350.83 -0.02%	14117.64 -0.01%		1019.62 -0.02%	
	mean	37270.21	12779.25	9352.39	14118.79		1019.78	
That must be the investment good?	meand	12834.53 -0.02%				12834.53 -0.02%		
	mean	12837.40				12837.40		

Note: In the example above, relative differences to the case with standard parameters, “mean” where added.

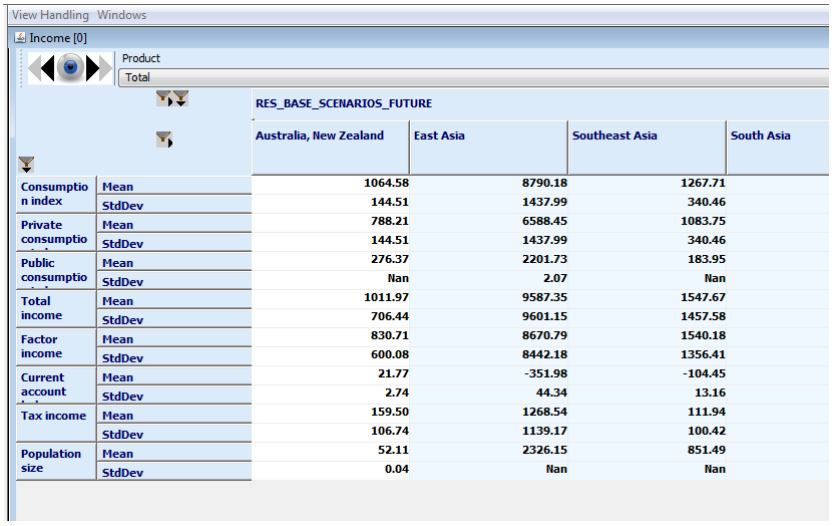
If the draws are selected, an additional dimension is introduced in most tables:



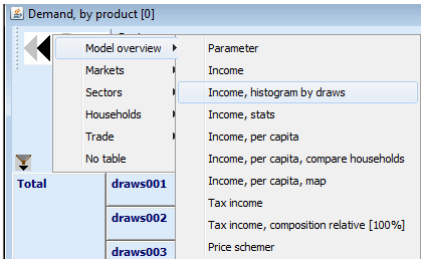
That allows a user to analyze each draw, if wished. Alternatively, the statistics built in the interface can be used. In order to do this, open with a right click in the table, the pop-up menu and select statistics:

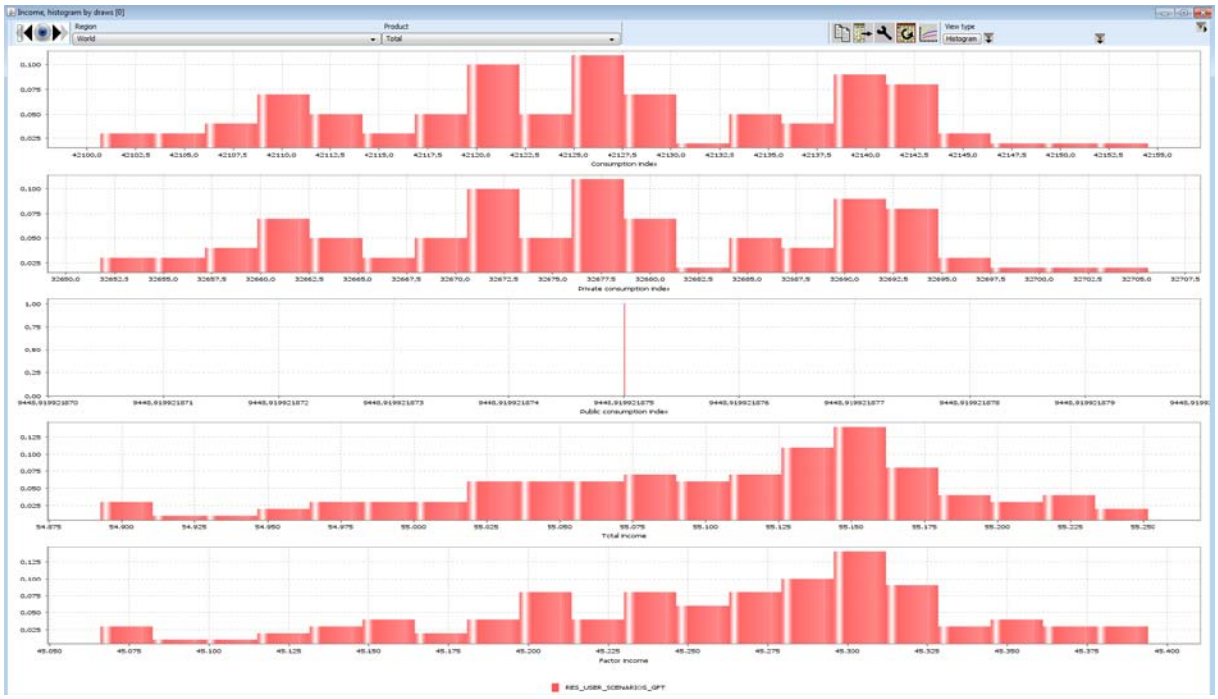


Select the statistics you want to be shown (such as mean and stdDev) and select “Only show outliers” to only show outliers and statistics:



Further work is necessary to add views which automatically generated e.g. histograms. Currently, only one view with the core income results is defined:

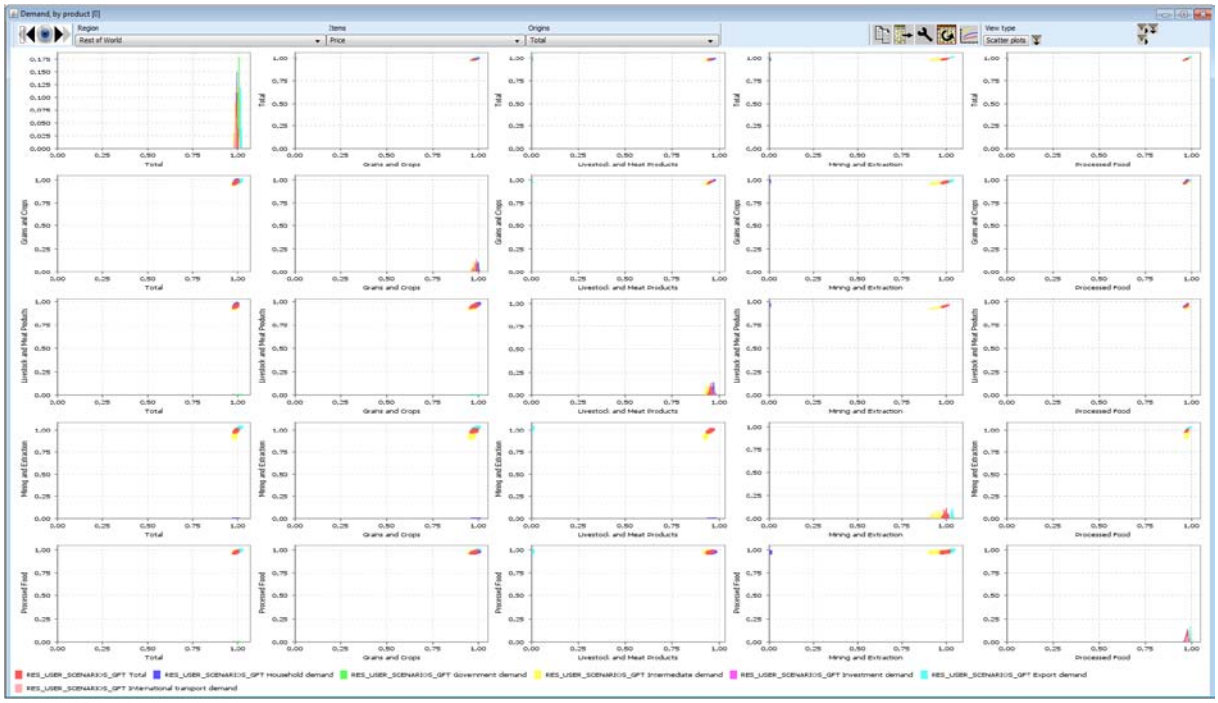




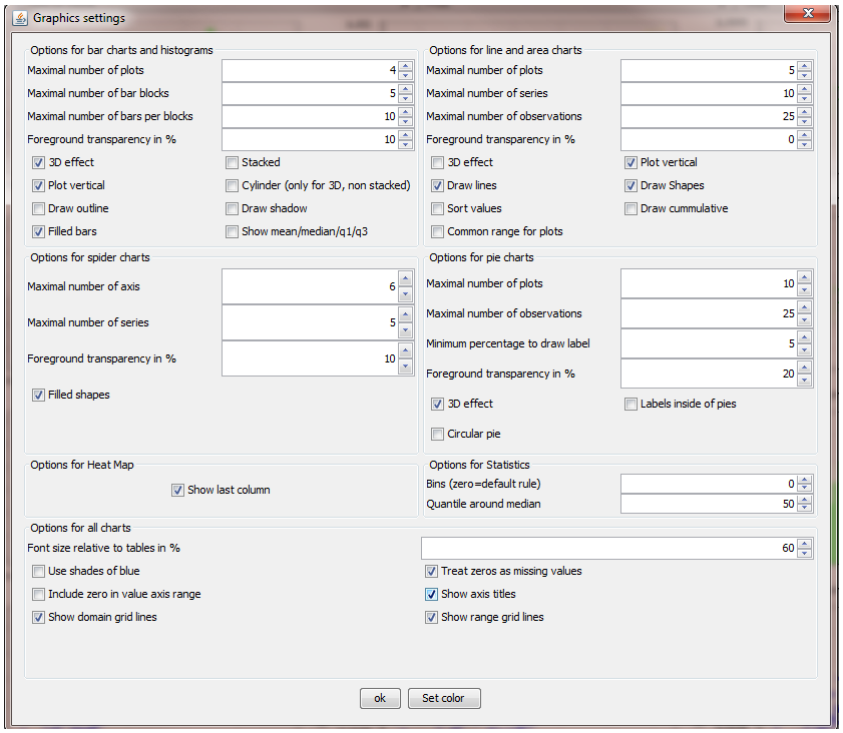
Note: If several scenarios are analyzed, the distributions for several scenarios will be shown together.

Alternatively, the “Scatter plot” type graphics can be used to plot additionally the correlation between the variables depicted:

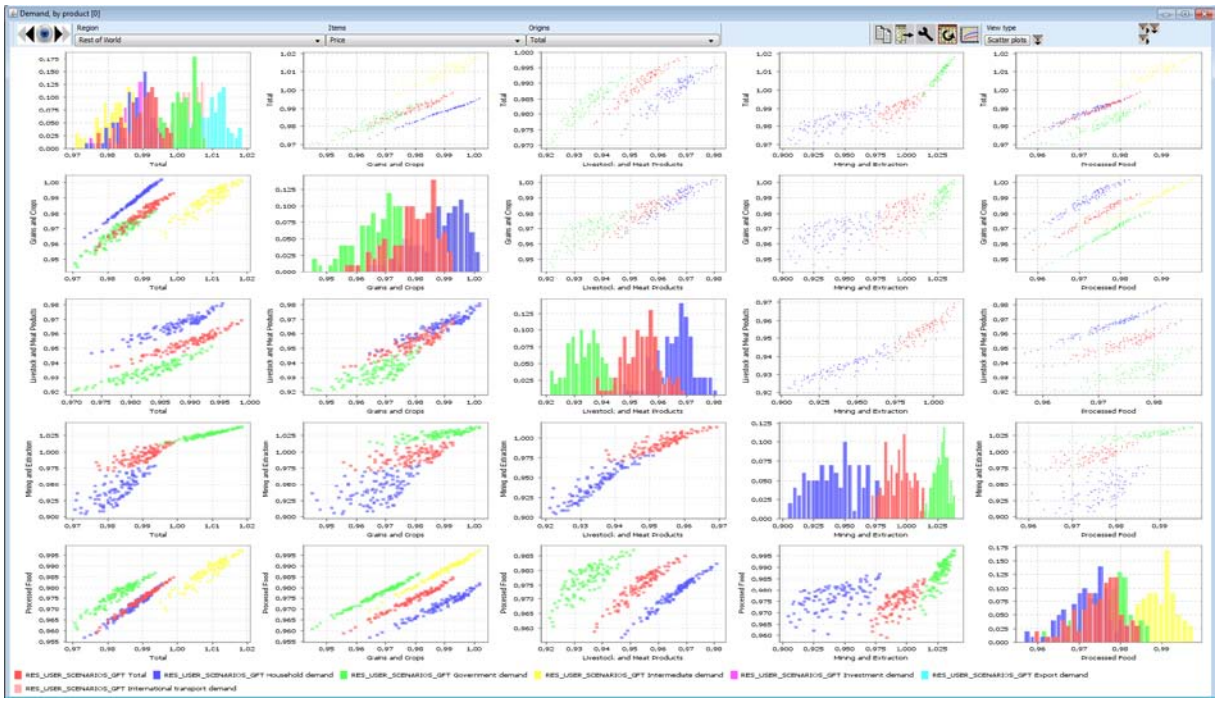
If the view type from existing tables is changed to “Scatter plots”, the result might look as below (the example shows different types of prices in each plot, and different sectors in the rows/columns:



The visualization of histograms and scatter plots can be improved, if the graphics settings are changed such that (1) zeros are treated as missing values, and (2) zero is not automatically included in the axis range:



The resulting scatter plots and histograms look as follows:



Note: If the entropy estimator is used, the "mean" calculated in the GAMS code uses the probabilities as weights, but they are not reflected in the GUI when the in-built statistics are used.

Getting an overview on the code

The GUI offers two possibilities to ease working with the code:

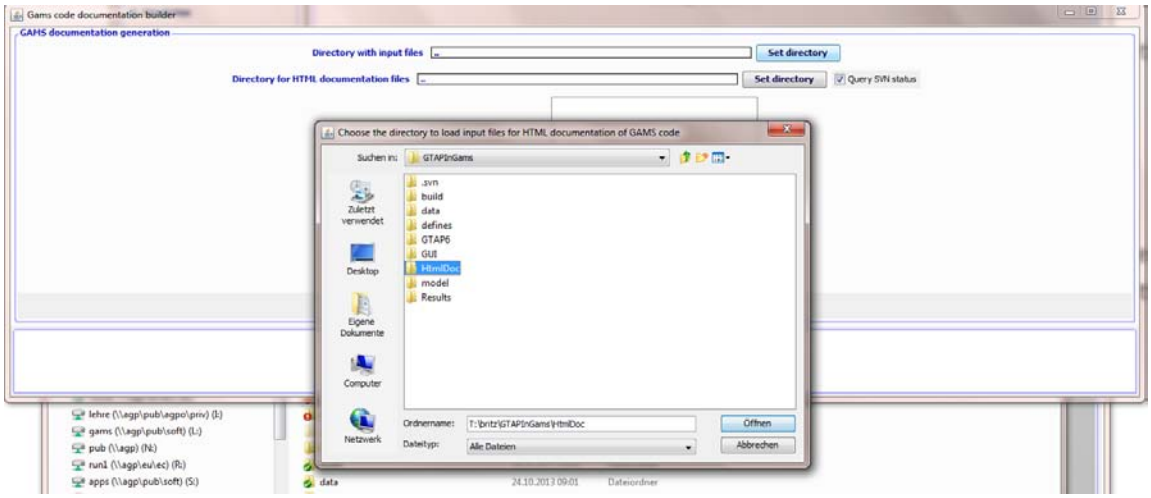
1. Generation of a set of static HTML pages
2. An equation, parameter and variable viewer linked to a specific model instance, which also allows looking at the GAMS code.

HTML documentation

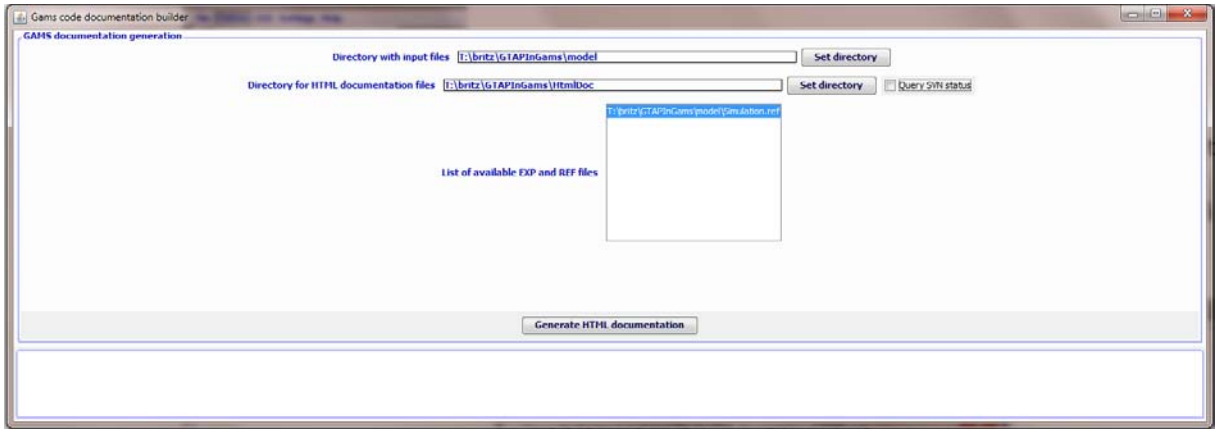
A HTML documentation of the code can be generated via "**Utilities/Build HTML documentation**":

The documentation will include any changes you introduced to the project's GAMS code.

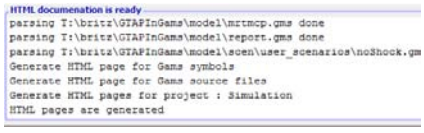
First choose the directory where the files should be stored (best use the "*HtmlDoc*" directory as shown below; beware: the utility will generate many files, do not use a directory where other files are already stored). The input files are found in the folder where GAMS was run, in the case of standard GTAP model in GAMS, typically in the "*model*" directory.



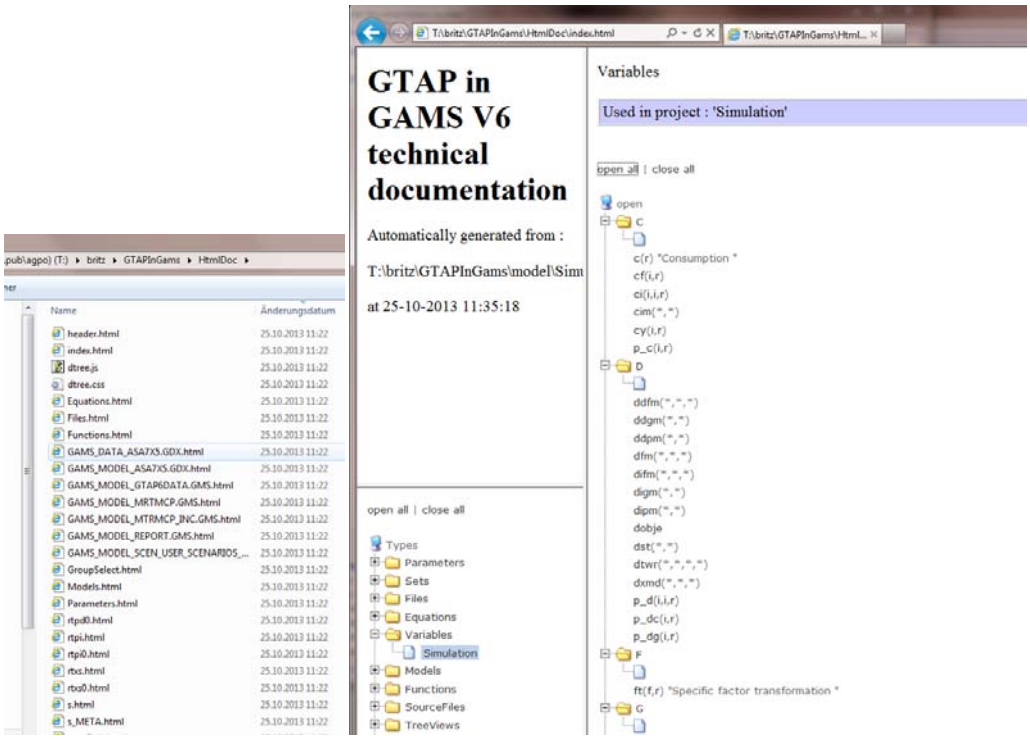
You should be able to find a file named “simulation.ref”. Make sure that “Query SVN Status” is switched off and next press the “Generate HTML documentation” button.



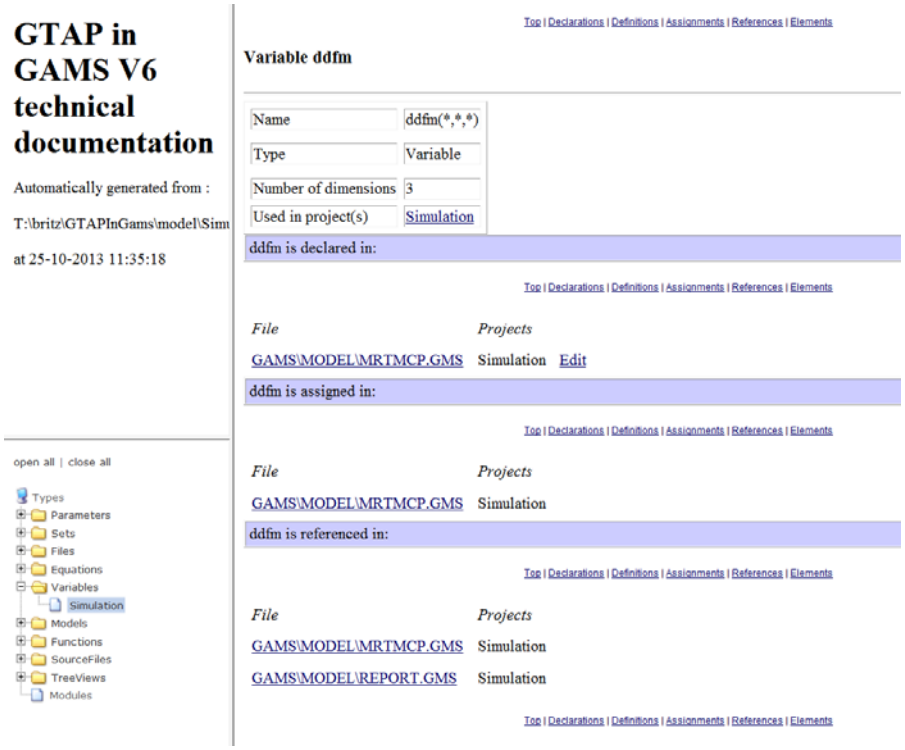
The program will work for a while and should end with “HTML documentation is ready” as shown below.


HTML documenation is ready
parsing T:\britz\GTAPInGams\model\mrtmcp.gms done
parsing T:\britz\GTAPInGams\model\report.gms done
parsing T:\britz\GTAPInGams\model\scen\user_scenarios\noShock.gm
Generate HTML page for Gams symbols
Generate HTML page for Gams source files
Generate HTML pages for project : Simulation
HTML pages are generated

Afterwards, you find a in the chosen output directory a list of file, “*index.html*” is the starting point.

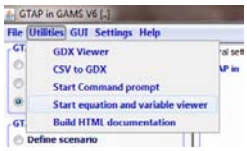


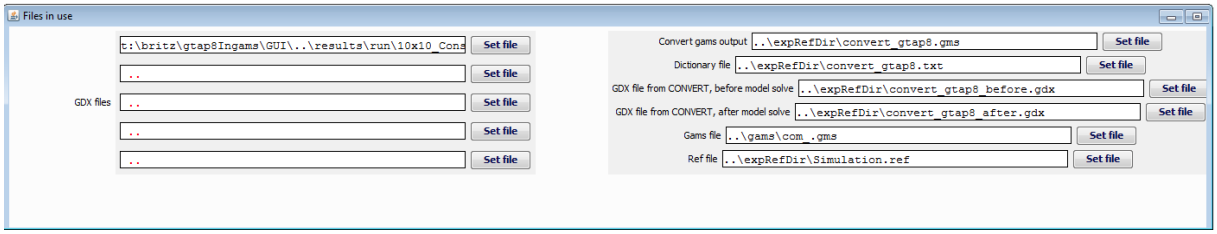
You can e.g. open all “Variables” used in the task “simulation”, clicking with the mouse opens a symbol page:



Equation and variable viewer (similar AnalyzeGE)

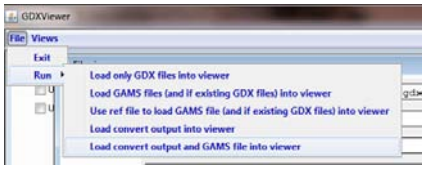
Whereas the HTML pages document the project, the equation and variable viewer helps you to analyze a specific model instance. The necessary input files are automatically generated with each run in comparative static mode if “CONVERT” output is chosen.



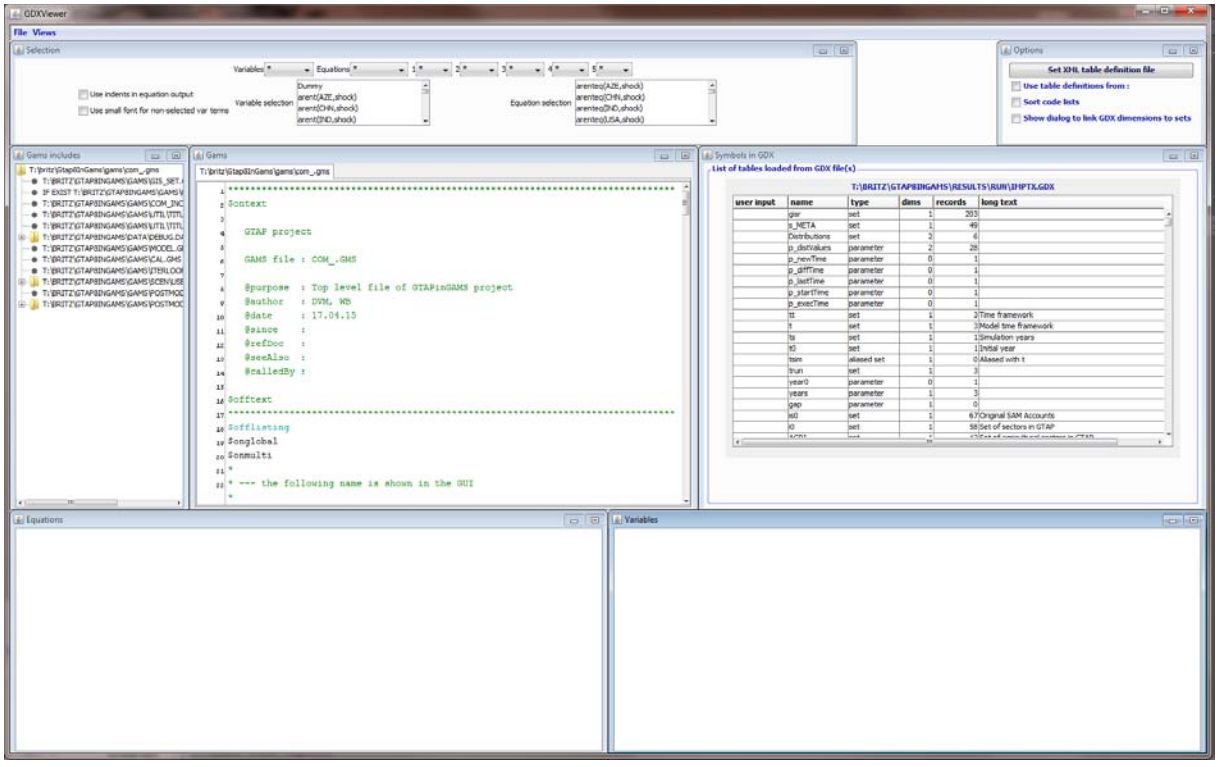


(note: one might want to change the first GDX files to the last experiment you have run)

Select “Load convert output and GAMS file into viewer” from the file menu:


GDXViewer
File Views
Exit
Run
Load only GDX files into viewer
Load GAMS files (and if existing GDX files) into viewer
Use ref file to load GAMS file (and if existing GDX files) into viewer
Load convert output into viewer
Load convert output and GAMS file into viewer

You get now view like below:



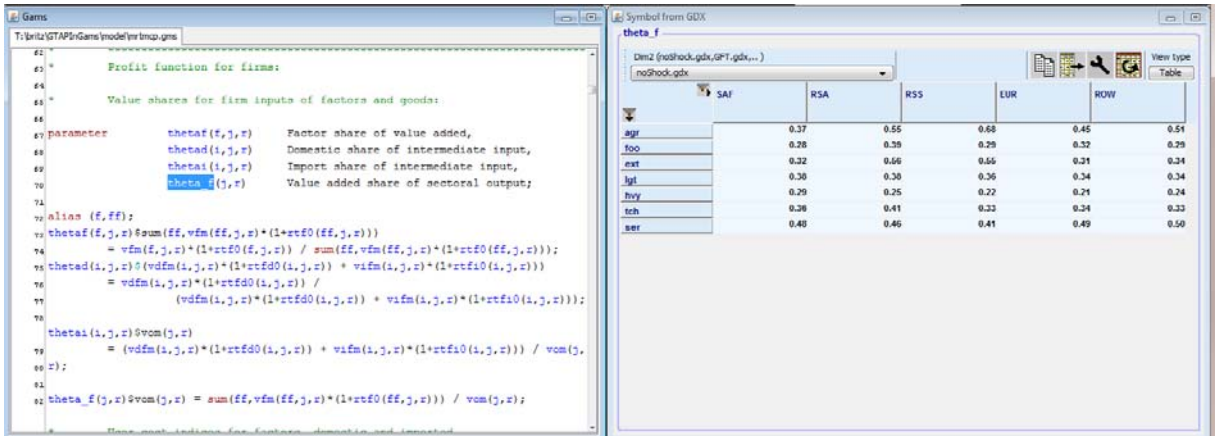
The viewer allows you to see:

- Linearized views on the equations, which you can select via the selection button above. The information in green behind each equation provides, separated by commata:
    1. The absolute change in the variable
    2. The absolute change multiplied with the Jacobian, i.e. the approximate change in the equation due to the simulated change

3. The contribution of the variable to the change in the LHS in percent, using information in the absolute changes in the LHS and the variable and the related Jacobian entries. The variable which the code analyzer has detected as the LHS is indicated with a * - in the example below that is xd. For the LHS, the relative change is report. In example below, the change is almost entirely coming from the change in nd (100%), the contribution of the other changes is very small.

```
xdeq(SouthAsia,MeatLstk-c,ProcFood-a,shock) "Agents demand for domestic goods " ..

  -0.00902671001153436*(0.9999999486802*(0.995354810037336*(1.0000000513198*ps(SouthAsia,MeatLstk-c,shoc
  {-0.005,-7.2E-5,1.0%})**(-2.13050842285156)
  +0.00464540767085857*(1.00000000249911*pmt(SouthAsia,MeatLstk-c,shock){0.0033,-4.7E-5,0.8%})**(-2.130
  50842285156))**(-0.469371530886303)/ps(SouthAsia,MeatLstk-c,shock){-0.005,-7.2E-5,1.0%})**3.130508422
  85156*nd(SouthAsia,ProcFood-a,shock){0.67,-0.006,100.0%}
  +0.183822553739227*xd(SouthAsia,MeatLstk-c,ProcFood-a,shock){0.034,0.0062,0.6% *}=E=0;
```

- A GDX viewer which allows merging of symbols (from one or several files) for combined analysis.
- A quick view on any symbol: click on any symbol in the GAMS code shown in blue, and it will be loaded in the "Symbol from GDX" view. You can directly compare your counterfactual against the base (or any scenarios against each other which you loaded as GDX file).

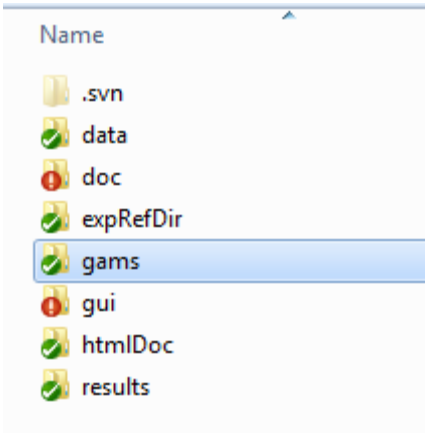


Technical documentation

Overview on directory structure

The screenshot below shows the directory structure of a "standard GTAP model in GAMS" installation. The information is only provided for those who have an interest into technical detail.

The GUI with the Java libraries is found in GUI. The results directory comprises the output from model runs. The data directory comprises also the GAMS readable versions of available data bases. All GAMS files can be found under "*gams*", in the subdirectory "*scen*", scenario templates and user defined scenarios are stored. Results of model runs are found under "*results/run*". The directory "*expRefDir*" and "*HtmlDoc*" are used to generate a HTML based documentation of the GAMS project.

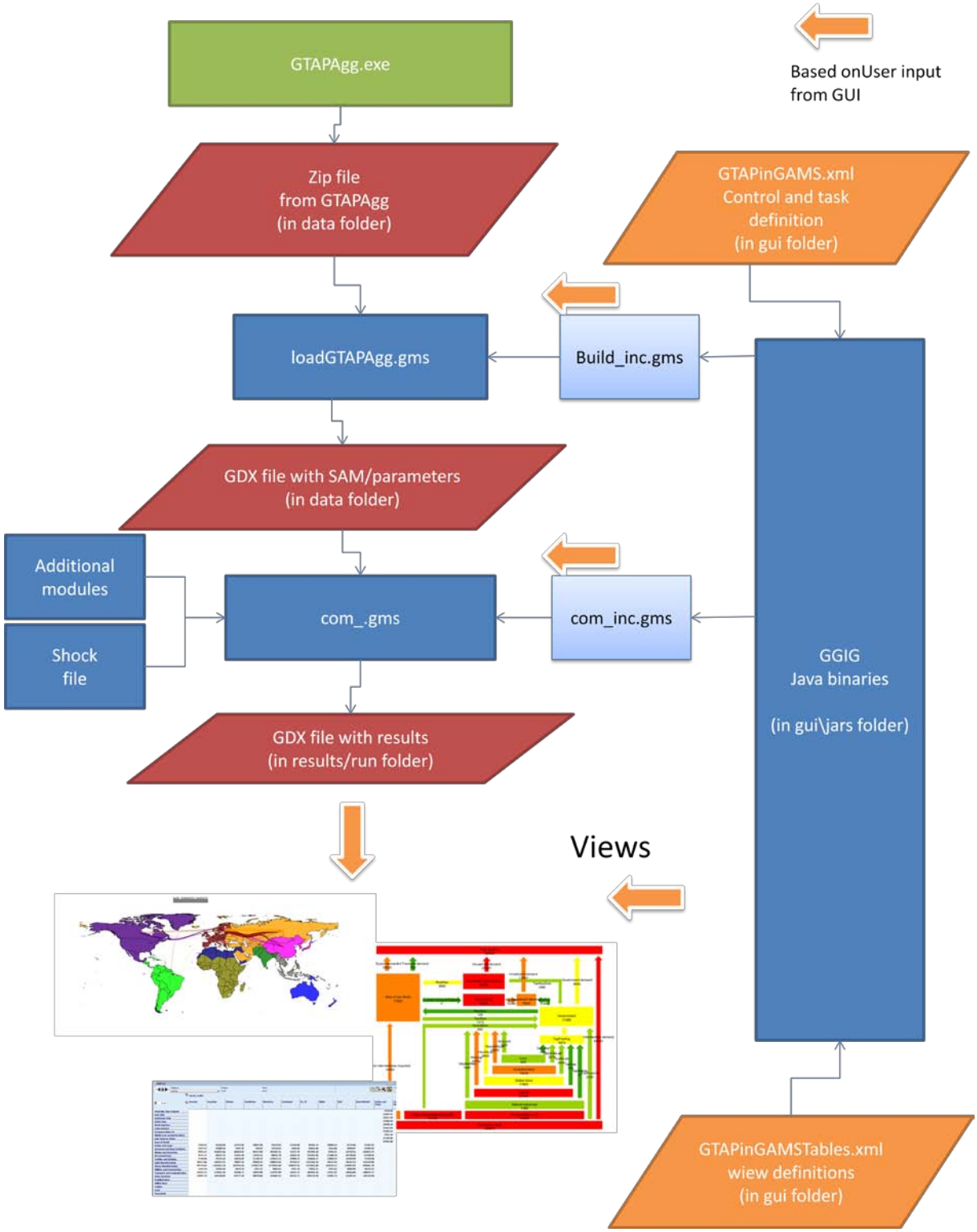


GUI overview

The GUI is programmed in Java and based on GGIG (Gams Graphical Interface Generator), which is also used for other projects, e.g. CAPRI (www.capri-model.org). While binaries are distributed for free, the Java code itself is currently not open source. It uses third-party libraries provided under licenses which allow for further distribution, and often as well code changes. The main jar to start is "*jars\ggig.jar*". In order to modify the GUI, no Java programming is necessary, as the Java code reads a XML file ("*gui\GtapInGams.xml*") which defines tasks, work steps and the individual GUI controls. Equally, a XML file ("*gui\GtapInGamsTables.xml*") defines the different views. The link to GDX files is based on the Java API distributed by GAMS.com as part of any GAMS installation, and uses dynamic link libraries. As both Java and GAMS are available also for non-Windows operation systems, it would be possible to also port the standard GTAP model in GAMS to other software platforms. Indeed, a CAPRI user had a few years back generated a native MAC version.

The interaction between the GUI and the GAMS processes is based on includes files which are generated anew for each run and capture the state of the user operated control.

Graphic: Overview on interaction between user, GUI and GAMS



Processes

In order to work with the model, GTAPAGG (which is clearly not part of the installation) must first be used to produce a zip file which comprises the necessary GTAP data, parameters etc. at the user chosen aggregation with regard to sectors, regions and factors. The GTAPAGG output should be stored in the “data” directory. The zip file comprises data in proprietary GEMPACK formats and cannot be used directly with GAMS code.

Next, the "LoadGTapAgg*.gms*" programs reads that output from GTAPAGG and converts it into GAMS symbols which are stored in a GDX container. It uses the "*ConvertGtapAgg.jar*" to read the content of the "*agg*" file with the aggregation information used by GTAPAgg to generate definitions with long texts which are temporarily outputted into "*agg.gdx*"..

On demand, the "filter.gms" program processes the inputted data to remove small values. The result from these step are stored in the "*data*" directory.

The GDX files generated from "loadGTAPAgg*.gms*" provide the input to actual model runs with "com_*.gms*". The GAMS code initializes all variables to benchmark values and starts the solver. That model start should basically need no iterations and lead to zero infeasibilities, it should prove that the model's parameters are correctly set up to replicate the benchmark. Next, the user chosen shock file is read and changes to parameters introduced. Afterwards, the model is solved for the shock.

Post-model processing

While the processes described above are more or less identical to the original code of Dominique and Van der Mensbrugghe and Tom Rutherford, the post-model processing was added to allow using the exploitation part of GGIG. The file "*report.gms*" stores the results back into a SAM like structure, e.g. consumer prices

```
p_results(r,"P",i,"hou","dom")          = p_dc.l(i,r);
```

The "*p_results*" parameter is stored in a gdx file and later read by the interface for exploitation. It is generally organized as follows:

- First dimension: regions, including a world aggregated stored under "wor"
- Second dimension: tables for values "V", quantities "Q", prices "P", tax rate "T" and tax income "G"
- Third dimension: commodities, factors
- Fourth dimension: institutions, sectors
- Fifth dimension: origins / destinations
- Sixth dimension: base, check or shock

Based on the flexible aggregation used in GTAP, the sets used for regions, sectors and factors are run specific and depend on the data set loaded.

The information about how the output is logically structured is also inputted in the GGIG definition file "*gtapInGams.xml*":

```
<task>
        <name>Simulation</name>
        <gamsFile>mrtmcp</gamsFile>
        <incFile>model\mtrmcp_inc</incFile>
        <curDir>model</curDir>

        <regionDim>0</regionDim>
        <dim5Dim>1,Items</dim5Dim>
        <productDim>2</productDim>
        <activityDim>3,Sectors and institutions</activityDim>
        <dim6Dim>4,Origins</dim6Dim>
        <scenDim>5,Scenarios</scenDim>

        <resdir>run</resdir>
        <gdxSymbol>p_results</gdxSymbol>
        <filemask>.*gdx$</filemask>
</task>
```

The “report.gms” program also performs aggregations, e.g. to total use

```
*
*   --- aggregate over origins (domestic and imported) to totals
*
    set agg(PQV) / Q,V,G /;


*
*   --- aggregate to total use
*
    set useCols / set.i,hou,gov,intTrs,set.s,inv /;

    p_results(r,agg,i,"use",oris)        = sum(useCols, p_results(r,agg,i,useCols,oris));
```

Average prices and tax rates are calculated afterwards:

```
p_results(r,"P",i,"use",oris) $ p_results(r,"Q",i,"use",oris)
   = p_results(r,"V",i,"use",oris) / p_results(r,"Q",i,"use",oris);

p_results(r,"T",i,"use",oris) $ p_results(r,"V",i,"use",oris)
   = p_results(r,"G",i,"use",oris) / p_results(r,"V",i,"use",oris);
```

The “*report.gms*” programs also stores meta-information on the run on the cube, such as the number of sectors, regions and factors; model size and solution status, and about how factor mobility is modeled:

```
* ------------------------------------------------------------------------------
*
*   Meta information on model structure and solve
*
* ------------------------------------------------------------------------------
*
   p_results("Wor","Meta","# of sectors","tots","dom")    = card(i);
   p_results("Wor","Meta","# of factors","tots","dom")    = card(f);
   p_results("Wor","Meta","# of regions","tots","dom")    = card(r);

   acronym MCP,CNS;
$iftheni %modeltype% == MCP

   p_results("Wor","Meta","Model type","tots","dom")    = MCP;
$else

   p_results("Wor","Meta","Model type","tots","dom")    = CNS;
$endif

$iftheni NOT %modeltye% == NONE


   p_results("Wor","Meta","# of equations","tots","dom") = %model%.numequ;
   p_results("Wor","Meta","# of variables","tots","dom") = %model%.numvar;
   p_results("Wor","Meta","SumInfes","tots","dom")       = %model%.sumInfes $ (%model%.sumInfes ne inf);
   p_results("Wor","Meta","NumInfes","tots","dom")       = %model%.numInfes $ (%model%.numInfes ne inf);
   p_results("Wor","Meta","# Iterations","tots","dom")                  = %model%.iterusd;
   p_results("Wor","Meta","# seconds solution time","tots","dom")       = %model%.resusd;

   acronym sluggish, mobile,fixed priced;


   p_results("Wor","Meta","Lab Market"      ,"tots","dom")       = %labmarket%    ;
   p_results("Wor","Meta","skl Market"      ,"tots","dom")       = %sklMarket%    ;
   p_results("Wor","Meta","capital Market"  ,"tots","dom")       = %capitalMarket%;
   p_results("Wor","Meta","res Market"      ,"tots","dom")       = %resMarket%    ;
   p_results("Wor","Meta","lnd Market"      ,"tots","dom")       = %lndMarket%    ;

$endif
```

That approach differs considerably from the way “runGTAP” allows to exploit results. For a formal discussion on these differences, see Britz 2014 and Britz et al. 2014. The “equation and variable” viewer, discussed above, allows views on the variables, equations and related parameters, more similar to the “runGTAP” exploitation tools.

Exploitation and flexible aggregation

The exploitation is based on views in the multi-dimensional cube defined in *p_results*. Several such cubes representing different counter-factual runs (plus typically a benchmark) can be loaded simultaneously in the viewer. The views are defined in “*GtapInGamsTables.xml*”. As the set of sectors, regions and factors can differ from run to run, “*report.gms*” generates a XML file “*generated.xml*” which that information, e.g. for the products:

```
   loop(i,
        put "<product>" /;
        put "   <itemName>", i.te(i),"</itemName>" /;
        put "   <key>", i.tl,"</key>" /;
        put "   <sel>all,products,sectorView</sel>" /;
        put "</product>" /;
   );
```

That file is included into the view definitions at run time.

In order to allow the viewer to work with maps, a co-ordinate set of individual countries is stored in “*GUI\world.zip*”. The standard case of GGIG applications are fixed lists of regional code and matching coordinate set which link each regional id to a list of polygons. The flexible regional aggregation in GTAP required a more flexible approach. The XML-definition of a view can register for a region a list of components under the tag “<disagg>”:

```
<region>
   <itemName>Australia, New Zealand</itemName>
   <key>Oceania</key>
   <sel>all</sel>
   <disagg>[AUS,FJI,PYF,KIR,NCL,NZL,MNP,PNG,SLB,TON,VUT,WSM,]</disagg>
</region>
```

The individual code listed under "<disagg>" indicate the regional Ids used in the coordinate set. In the case of GTAP, these are codes for individual countries. The actual mapping between the aggregates used in the current model instance and the GTAP regions is read from an "*agg*"-file; the mapping between the GTAP regions code in the data base and the individual country ids is defined in "*model\map_regions.gms*", e.g.

```
rrmap("BRN","XSE") = YES;
rrmap("MMR","XSE") = YES;
```

That file should be currently set-up to work with GTAP version 8. Other versions require an update of the mappings, as the list of GTAP regions might differ.

A first view reports the meta-information on the model. It is here used as a first example for the XML based definition of views. Basically, each view can define filters in the different dimension. The filters end with "sel" and start with the logical name of a dimension. To give an example: where normally information on a sector or institution would be stored (fourth dimension, the "activityDim"), the label "tots" is used. It is chosen in the filter below based on a regex expression:

```
<activitySel>tots_REGEX</activitySel>
```

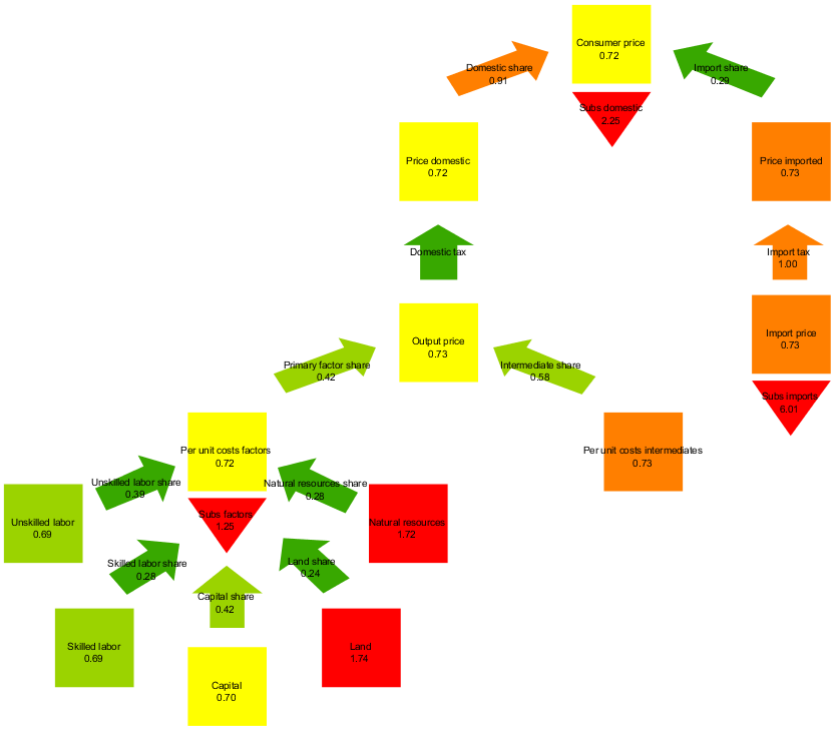
```
<table>
   <theme>Model overview</theme>
   <name>Model properties</name>
   <regionSel>Wor_REGEX</regionSel>
   <regionText>hide</regionText>
   <productSel>.*_REGEX</productSel>
   <productText>hide</productText>
   <activitySel>tots_REGEX</activitySel>
   <activityText>hide</activityText>
   <dim5Sel>Meta</dim5Sel>
   <dim5Text>hide</dim5Text>
   <hideEmptyRows>YES</hideEmptyRows>

   <dim6Text>hide</dim6Text>

   <defview>Table</defview>
   <defpivot>0P0S</defpivot>

   <itemDim>dim6</itemDim>
      <item>
           <itemName>Dummy</itemName>
           <key>dom</key>
      </item>
</table>
```

Another example provides the table for intermediate demand. Here all products and sectors are selected, instead of a regex expression, the pre-defined lists found in "generated.xml" are used:

```
<table>
   <name>Intermediate demand</name>
   <theme>Markets</theme>
   <regionSel>all</regionSel>
   <productSel>products</productSel>
   <productText>Inputs</productText>
   <activitySel>sectors</activitySel>
   <activityText>sectors</activityText>
   <dim5Sel>all</dim5Sel>
   <dim6Text>Origin</dim6Text>
   <dim6Sel>origins</dim6Sel>

   <defview>Table</defview>
   <defpivot>0PSA</defpivot>
   <itemDim>dim6</itemDim>
      <item>
           <itemName>Total</itemName>
           <key>tot</key>
      </item>
      <item>
           <itemName>Imported</itemName>
           <key>Imp</key>
      </item>
      <item>
           <itemName>Domestic</itemName>
           <key>Dom</key>
      </item>
</table>
```

Details on how views can be defined can be found in GGIG programming guide, Britz 2010b.

As part of the exploitation tool, the composition of the producer price (= per unit costs) and the consumer prices is visualized as shown below:



That type of schemer is constructed based on registering hyperlinks to graphical symbols on a powerpoint slide and storing the presentation as html. A java programs reads the coordinates of the mouse regions which linked to the hyperlinks and stores them technically as if they would be coordinates for a geographic map. The hyperlinks registered in powerpoint must hence match the keys in a data dimension put in the row dimension of a tabular view, the following screenshots show a part of the view definition underlying the scheme visualize above and related GAMS statements:

```
   <item>
      <key>pc</key>
      <itemName>Consumer price</itemName>
   </item>
   <item>
      <key>esubd</key>
      <itemName>Subs domestic</itemName>
   </item>
   <item>
      <key>pic</key>
      <itemName>Price imported</itemName>
   </item>
   <item>
      <key>pics</key>
      <itemName>Import share</itemName>
   </item>
```

```
* ------------------------------------------------------------------------------
*
*   Store prices for price schemer
*
* ------------------------------------------------------------------------------

    p_results(r,"P",i,"if","tot")      = p_results(r,"P","int",i,"tot");
    p_results(r,"P",i,"cf","tot")      = cf.l(i,r);
    p_results(r,"P",i,"py","tot")      = p_results(r,"P","out",i,"dom");
    p_results(r,"P",i,"pc","tot")      = p_results(r,"P",i,"hou","tot");
```

References

Britz, W. (2010a): GAMS Graphical Interface Generator User Guide. Institute for Food and Resource Economics, University Bonn. In: http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG_user_Guide.pdf.

Britz, W. (2010b): GAMS Graphical Interface Generator Programming guide. Institute for Food and Resource Economics, University Bonn. In: http://www.ilr.uni-bonn.de/agpo/staff/britz/GGIG_programming_guide.pdf

Britz, W. (2014): A New Graphical User Interface Generator for Economic Models and its Comparison to Existing Approaches, German Journal for Agricultural Eonomics, 63(4), forthcoming

Britz, W., Domínguez, I. P. and Gopalakrishnan, B. N. (2015): Analyzing results from agricultural large-scale Economic Simulation Model: Recent Progress and the Way Ahead, German Journal for Agricultural Economics, 107-119

Britz, W., Kallrath, J. (2012): Economic Simulation Models in Agricultural Economics: The Current and Possible Future Role of Algebraic Modeling Languages, in: Kallrath, J. (eds.): Algebraic Modelling Systems: Modeling and Soving Real World Optimization Problems: 199-212, Springer, Heidelberg, Germany

Domínguez, I. P., Britz, W., & Gopalakrishnan, B. N. Post-model Analysis in large-scale models: the examples of Aglink-Cosimo, CAPRI and GTAP, Paper presented at 2012 GTAP conference.

Rutherford T. and Harbor A. (2005). GTAP6inGAMS: The Dataset and Static Model. Prepared for the Workshop: “Applied General Equilibrium Modeling for Trade Policy Analysis in Russia and the CIS", The World Bank Resident Mission, Moscow, December 1-9, 2005, http://www.mpsge.org/gtap6/gtap6gams.pdf

Arndt, Channing. "An introduction to systematic sensitivity analysis via Gaussian quadrature." GTAP Technical Papers (1996): 2., https://www.gtap.agecon.purdue.edu/resources/download/1209.pdf

Van der Mensb…

Ko, J.-H. and Britz, W. (2013) Does Regional and Sectoral Aggregation Matter? Sensitivity Analysis in the Context of an EU-South Korea FTA", https://www.gtap.agecon.purdue.edu/resources/download/6313.pdf